

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-149599

(43)Date of publication of application : 27.05.1994

(51)Int.Cl.

G06F 9/46

(21)Application number : 04-327449

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 12.11.1992

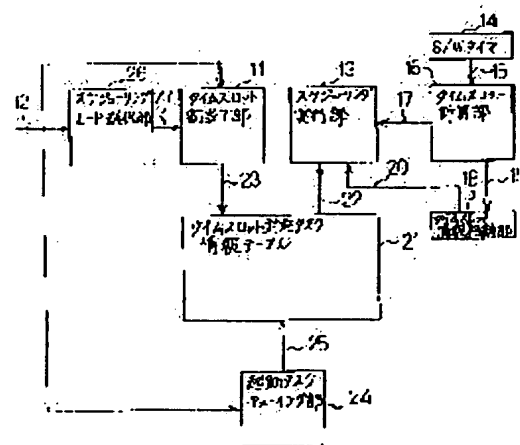
(72)Inventor : NAKAMURA MASAHIRO

(54) COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To provide the computer system which can schedule tasks of all periods by providing scheduling in which case conventional periodical scheduling cannot be scheduled and there are tasks having execution periods shorter than a minimum reference period.

CONSTITUTION: This computer system consists of a scheduling mode selecting part 26 and a time slot assigning part 11 in addition to the system constituting the conventional scheduling system. The scheduling mode selecting part 26 changes the scheduling mode from the periodical scheduling mode to the priority when there are tasks existed that are execution periods shorter than the minimum reference period which becomes the reference period at the time of periodically checking the activation of tasks by a scheduling execution part 13 in the activation requesting tasks. The time slot assigning part 11 generates table information 23 of scheduling in accordance with the selected scheduling mode.



LEGAL STATUS

[Date of request for examination] 28.09.1995

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2760721

[Date of registration] 20.03.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

SPECIFICATION <EXCERPT>

[0018] FIG. 2 is a functional block diagram showing an internal structure of the above-mentioned scheduling mode selecting unit 26. In FIG. 2, a reference numeral 26a is an execution period determining unit which determines an execution period of each task based on task information 12. A reference numeral 26b is a classifying unit which classifies each task into one of execution levels ranging from a level with a short execution period to a level with a long execution period based on the result of the determination by the execution period determining unit 26a. A reference numeral 26c is a comparing unit which judges whether or not an execution period of an execution level, into which a task with a minimum execution period is classified by the classifying unit 26b, is shorter than the above-mentioned minimum reference period. A reference numeral 26d is a scheduling mode switching unit which switches between a conventional periodical scheduling mode and a priority scheduling mode, which is a new scheduling mode, based on the result of the comparison. The priority scheduling mode assigns an execution level, into which a task with a minimum execution period is classified, to every time slot. At that time, an output from the scheduling mode switching unit 26d is a scheduling mode signal 27. In addition, the execution period determining unit 26a and the classifying unit 26b correspond to an execution period determining unit 11a and a classifying unit 11b, respectively, which are originally included in a time slot assigning unit 11 (see FIG.17). Therefore, it is also possible to have only one set of these so as to be shared by

the time slot assigning unit 11 and the scheduling mode selecting unit 26.

[0019] Next, the time slot assigning method according to the present embodiment is explained with flow charts FIG.3 and FIG.4. First, the scheduling mode selecting unit 26 determines execution periods of all active tasks by using the execution period determining unit 26a, and classifies the active tasks according to their execution periods into the execution levels by using the classifying unit 26b (ST31). For example, a task with an execution period 50 ms is classified into an execution level 1, and a task with execution period 70 ms is classified into an execution level 2. Next, the scheduling mode selecting unit 26 compares the minimum reference period and the execution period in the execution level 1, to which the task with the minimum execution period belongs, by using the comparing unit 26c (ST32). If the execution period in the execution level 1 is smaller, the priority scheduling mode is selected by the scheduling mode switching unit 26d for switching (ST34). On the other hand, if it is larger, the conventional scheduling mode is selected for switching (ST33). Next, the time slot assigning unit 11 receives the scheduling mode signal 27 and checks its mode (ST35). In the case of the priority scheduling mode, as shown in FIG. 5, the time slot assigning unit 11 assigns the execution level 1 to all of the time slots 1 to N in a time slot managing table 211 (ST36). In the case of the periodical scheduling mode, a processing is performed in the same way as the conventional system. That is, the time slot assigning unit 11 performs the following processing with respect to execution levels 1 to M (ST37). As for the execution level 1, multiplying the sum of all of the task execution time in the execution level 1 by the ratio of the execution period in the execution level 1 to the minimum reference period gives the time which the total execution time of the tasks in the execution level 1 takes in the minimum reference period (ST38). As described in the

conventional example, for example, in the case where an execution period of tasks in the execution level 1 is 100 ms, the total execution time in the execution level 1 is 40 ms, and the minimum reference period is 50 ms, $40 \times (50/100) = 20$ ms is the obtained value. That is, the execution level 1 requires an execution time for 20 ms within the minimum reference period. By dividing the obtained value by a time-slot time and rounding up, the number of time slots required in the execution level 1 within the minimum reference period is obtained (ST39). In the above example, when time-slot time is 10 ms, $20 / 10 = 2$, which means two time slots are needed. The number of time slots required in each execution level is obtained by repeating the processing as described above from execution levels 2 to M (ST40). Next, the number of time slots required for the execution time of the system which is needed for executing a task (ST41). The number of time slots required in each execution level is calculated with the above procedure. However, when the sum of the number of time slots exceeds the number N of the time slots in the minimum reference period, the maximum execution load is exceeded, which results in losing control. In order to avoid this, the sum is compared to N (ST42), then if the sum is over the number of time slots N, time slots just as many as needed is stored in the time slot managing table 211 sequentially from the execution level 1. Then, at the time when the table is filled, the remaining levels are ignored (ST44). When N is larger than the sum, the time slots as much as needed is stored in the time slot managing table 211 sequentially from the execution level 1 without any problems (ST43). The assignment of the execution levels to the time slots is completed as above.

[0020] A scheduling execution after the time slot assignment is performed in the same operation as the conventional way. In this regard, in the case where the time slot managing table 211 is generated in the priority scheduling mode as shown in FIG. 5, the

flow chart of FIG. 21 mentioned in the conventional example is changed to FIG. 6. Every time slot acquires execution level 1 (ST212), level managing tables 212a-212c are checked from the execution level 1 as shown in FIG. 22, and a queuing task found at first is executed. As a result, the maximum priority is given to the execution level 1 and the minimum priority is given to the execution level M. That is, because every time slot acquires the execution level 1, and a task to be executed is determined according to the result of a search for the execution level as shown in FIG. 22, a task in a higher execution level is not executed as long as a task in a lower execution level is not executed. Consequently, the priority scheduling, in which a task in smaller execution level is given a higher priority, is provided. In this case, FIG. 16 mentioned in the conventional example is changed to FIG. 7.

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-149599

(43)公開日 平成6年(1994)5月27日

(51)Int.Cl.⁵

G 0 6 F 9/46

識別記号

3 4 0 B 8120-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 3 (全 30 頁)

(21)出願番号

特願平4-327449

(22)出願日

平成4年(1992)11月12日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 中村 正博

神戸市兵庫区和田崎町1丁目1番2号 三

菱電機株式会社制御製作所内

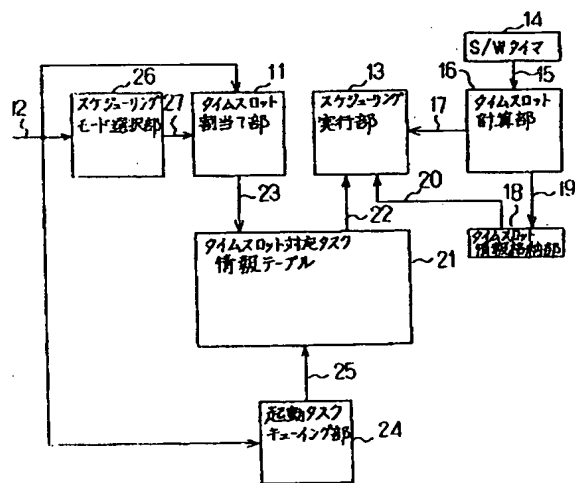
(74)代理人 弁理士 宮園 純一

(54)【発明の名称】 計算機システム

(57)【要約】

【目的】 従来の周期的スケジューリングでは不可能な、実行周期が最小基準周期未満のタスクが存在する場合のスケジューリングを提供し、全ての周期のタスクのスケジューリングを可能にした計算機システムを得ることを目的とする。

【構成】 従来のスケジューリング方式を構成するシステムに加えて、起動要求タスク中に、スケジューリング実行部13が周期的にタスクの起動をチェックする際の基準周期となる最小基準周期未満の実行周期のものが存在する場合、スケジューリングモードを周期的スケジューリングモードからプライオリティスケジューリングモードに変更するスケジューリングモード選択部26と、選択されたスケジューリングモードに従いスケジューリングのテーブル情報23を生成するタイムスロット割当て部11から構成される。



12 ; タスク情報
15 ; 定周期初値
17 ; タイムスロット変更期
19 ; タイムスロット値(変更値)
20 ; タイムスロット値(現在値)
22 ; テーブル情報(現在値)
23 ; テーブル情報(変更値)
25 ; 起動要求タスク
27 ; スケジューリングヘッド信号

【特許請求の範囲】

【請求項1】 固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当ててタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなかを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切替えるスケジューリングモード切替手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切替手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切替え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てることを特徴とする計算機システム。

【請求項2】 固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当ててタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段と、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理の優先度に基づき割込み処理が周期的タスクより優先度が高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させることを特徴とする計算機システム。

【請求項3】 固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有す

る計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当ててタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなかを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当ててプライオリティスケジューリングモードとを切替えるスケジューリングモード切替手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段と、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切替手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切替え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てることを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、複数タスクが実行される計算機システムに係わり、特に固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムに関するものである。

【0002】

【従来の技術】図15は、例えば「三菱電機統合制御システムMELTAS J P-EI S/W構築マニュアル」（1991年5月第1版発行）で提供されている計算機システムのスケジューリングに関する部分の構成図である。

【0003】まず、周期的スケジューリングの概念について、図16を用いて説明する。図中の最小基準周期とは、周期的にタスクの起動をチェックする際の基準周期となるもので、周期的に実行されるアプリケーションタ

スク1〜L (Lはタスク数) をスケジューリングする最小周期である。タイムスロットとは、上記最小基準周期を所定値N (ユーザがシステムに合わせて決定する) で割った時間単位で、タイムスロット時間毎に、スケジューラが実行タスク1〜Lの選択を行う。実行レベルとは、実行タスク1〜Lを周期毎に分類した複数のタスクの集合で、タイムスロット1〜Nにその実行レベル1〜Mが割当てられる (割当て方法は後述する)。スケジューラはスケジューリング実行時に、あるタイムスロットでそのタイムスロットに割当てられた該実行レベルの起動待ちタスクを優先的に実行する (実行レベルの選択方法は後述する)。

【0004】図16の例では、タイムスロット1とタイムスロット2に実行レベル1を、タイムスロット3に実行レベル2を、タイムスロット(N-1)に実行レベルMを割当て、タイムスロットNは、実行レベルの選択時間やタイムスロットの計算時間として、各タイムスロットに要するシステム時間のための予備のタイムスロット(α)であり、実行レベル1を割当てる。実行レベル1にはタスク1とタスク2の2つのタスクが、実行レベル2にはタスク3が、実行レベルMにはタスクLが所属している。スケジューリング時、タイマの値から、後述するタイムスロット計算部がタイムスロット値を計算する。タイムスロット1においては、実行レベル1が選択され、実行レベル1のタスク1が実行されている。タイムスロット2においても、実行レベル1が選択され、同様にタスク1が実行されている。タイムスロット3においては実行レベル2が選択され、実行レベル2の所属タスク3が実行される。タイムスロットNは、予備のタイムスロットなので、最小基準周期内で未実行のタスクが実行される。以上の様な実行レベル選択が、最小基準周期毎に繰り返される。

【0005】図15において、11は、ユーザから提供される各タスクに対する固有の実行周期と実行時間を含むタスク情報12からタスクを実行レベルに分類し、実行レベルを割当てたタイムスロットの情報を含むテーブル情報 (変更値) 23を生成し、タイムスロット対応タスク情報テーブル21に該テーブル情報23を格納するタイムスロット割当て部であり、図17に示すように、タスク情報12から各タスクの実行周期を判別する実行周期判別手段11aと、この判別結果に基づき各タスクを実行周期の短い実行レベルから長いレベルまでレベル分けする分類手段11bと、各タイムスロットに上記実行レベルを割当てるタイムスロット割当て手段11c等から成る。一方、図15の13は後述するタイムスロット計算部16からのタイムスロット変更割込み17をトリガとして、タイムスロット対応タスク情報テーブル21からのテーブル情報 (現在値) 22と、後述するタイムスロット情報格納部18からのタイムスロット値 (現在値) 20に基づき実行タスクを選択し、起動をかける

スケジューリング実行部、16はオペレーティングシステムの機能であるソフトウェア (以下、S/Wと略記する) タイマ14からの定周期割込み15で起動され、現在の時刻からタイムスロット値 (変更値) 19を算出し、タイムスロット情報格納部18にその値を書き込むとともに、このタイムスロット情報変更時にスケジューリング実行部13にタイムスロット変更割込み17をかけ、現在のタスク実行を中断させ、起動タスク選択をスケジューリング実行部13に行わせるタイムスロット計算部であり、算出したタイムスロット値は内部にも保持され、このタイムスロット値が次の算出時にタイムスロットの前回値としてインクリメントされる。24は、ユーザからタスク情報12として渡される起動要求タスク25を、タイムスロット対応タスク情報テーブル21内の後述するタスク管理テーブルで生成されるキュー (待ち行列) にキューイングする起動タスクキューイング部である。

【0006】図18は図15のタイムスロット対応タスク情報テーブル21に含まれる各テーブルの詳細図である。211はタイムスロット1〜Nの各々に割当てた実行レベル1〜Mの情報を格納するタイムスロット管理テーブル、212a〜212cは各タイムスロット値に割当てられた各実行レベルのタスクのキューイング状態を格納するレベル管理テーブル、213a1〜213an, 213b1, 213c1, 213c2は上述したキューを生成するためのタスク管理テーブルである。上記タイムスロット管理テーブル211には、各タイムスロット1〜Nに割当てられた実行レベル1〜Mに対応するレベル管理テーブル212a〜212cのアドレスを示すポインタa〜cが格納される。タイムスロットNはシステム時間のための予備のタイムスロット(α)で、実行レベル小のタスクを優先的に実行するために実行レベル1を割当てる。タスク管理テーブル213a1〜213c2には、次のタスク管理テーブルのアドレスを指すポインタ (ただし、キューの最後尾のタスク管理テーブル213an, 213b1, 213c2のポインタは後に続くテーブルがないことを示すNULLポインタ) と、起動タスクの情報が格納される。また、レベル管理テーブル212a〜212cには、各々の実行レベルでのキューの先頭にあるタスク管理テーブルのアドレスを指すポインタと、最後尾にあるタスク管理テーブルのアドレスを指すポインタが格納される。例えば、レベル1用のレベル管理テーブル212aには、実行レベル1に所属するキューイングタスクの先頭タスク管理テーブル213a1のアドレスを示すポインタa1と、同最後尾タスク管理テーブル213anのアドレスを示すポインタanを格納することで、これらのタスク管理テーブル213a1〜213anに示されたタスクが実行待ちタスクであるという情報を提供する。

【0007】次に動作について説明する。まず、タイム

スロット割当て部11による上記タイムスロット管理テーブル211への実行レベルの割当て手段について、図19、図20のフロー図を用いて説明する。割当てに先立って、入力されたタスク情報12に基づき、実行周期判別手段11aと分類手段11bにより、全ての起動タスクを実行周期毎にM個（ユーザが実行タスク数とシステムの能力から決定する数）の実行レベルに分類する

(ST191)。次に、タイムスロット割当て手段11cにより、実行レベル1～Mについて以下の処理を繰り返す(ST192)。まず、実行レベル1について、最小基準周期単位の実行時間の合計を、実行レベル1に所属する全てのタスク実行時間の総和に、実行レベル1の実行周期の最小基準周期に対する割合を掛けることで求める(ST193)。例えば、実行レベル1のタスクの実行周期が100ms、実行レベル1のタスクの実行時間合計が40ms、最小基準周期が50msの時、 $40 \times (50/100) = 20\text{ms}$ が求める値となる。即ち、実行レベル1は最小基準周期内で20msの実行時間が必要となる。その結果をタイムスロット時間で割り、小数点以下を切上げ、最小基準周期内で実行レベル1に必要なタイムスロット数を求める(ST194)。前述の例では、タイムスロット時間が10msの時、 $20/10 = 2$ 、故に2個のタイムスロットが必要となる。以上の処理を同様に実行レベル2からMまで繰り返し(ST195)、各実行レベルに必要なタイムスロット数を求める。次に最小基準周期内でタスクの実行に必要なシステムの実行時間に必要なタイムスロット数を、システム実行時間をタイムスロット時間で割り、小数点以下を切り上げて求める(ST196)。以上で、各々の実行レベルとシステムが必要とするタイムスロット数が算出できるが、ただし、その合計が最小基準周期のタイムスロット数Nを越えると、実行最大負荷を越えスケジューリング不可となるため、その合計と最小基準周期のタイムスロット数Nを比較し(ST197)、合計の方がタイムスロット数Nより大きい場合は、実行レベル1から順に必要なタイムスロット数だけ、レベル管理テーブルへのポインタをタイムスロット管理テーブル211に格納し、タイムスロット管理テーブル211が満たされたとき、残りのレベルは無視するようにする(ST199)。合計がタイムスロット数Nより小さい場合は、実行レベル1から順にタイムスロット管理テーブル211に格納する(ST198)。以上でタイムスロットへの実行レベルの割当てを終了する。

【0008】次に、スケジューリング実行部13によるタイムスロット管理テーブル211を用いたスケジューリング方法を、図21のフロー図を用いて説明する。S/Wタイマ14からの定周期タイマ割込み15を受けて、タイムスロット計算部16はタイムスロット値19を計算し、タイムスロット情報格納部18にその値を格納し、タイムスロット変更割込み17をスケジューリン

グ実行部13に出力する。スケジューリング実行部13はこの変更割込み17を受けて、動作を開始する。現在のタイムスロット値20をタイムスロット情報格納部18から獲得し(ST211)、獲得したタイムスロット値に割当てられている実行レベルを図18のタイムスロット管理テーブル211の実行レベル1～Mから獲得する(ST212)。該実行レベルにキューイングされたタスク（実行待ちタスク）の存在を、該レベル管理テーブル212a～212cを用いてチェックし(ST213)、キューイングされていればその先頭タスクを実行する(ST214)。例えば、タイムスロット情報格納部18からのタイムスロット値が1の時、スケジューリング実行部13はタイムスロット管理テーブル211によりポインタaで示されるレベル管理テーブル212aを確認し、そのポインタa1で示される先頭タスク管理テーブル213a1のタスクを実行する。キューイングされていなければ実行レベルをインクリメントし(ST216)、次の実行レベルのレベル管理テーブルについてキューイング状態のチェックを繰り返す。実行レベルのチェックは、現タイムスロットに割り当てられた実行レベルを先頭に、図22のようにサイクリックに行うため、実行レベルがMを越えたら実行レベルを1に戻す(ST217, ST218)。以上のチェックで全ての実行レベルを一巡したら(ST219)、全ての実行レベルに実行タスク無しと判断する(ST220)。実行タスクが有る場合、無い場合ともに、タイムスロット変更割込み17を待って(ST215)、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。

【0009】

【発明が解決しようとする課題】従来の計算機システムは以上の様に構成されているので、図19、図20に示したタイムスロット割当て方法では、周期的スケジューリングを行なうことでプロセッサへの負担を均等にできるが、最小基準周期でタスク起動の周期的チェックを実行するため、最小基準周期未満の周期的スケジューリングは不可能となる。このため、最小基準周期未満の実行周期のタスクについては、このようなタスクが存在しないようにユーザは実行周期を決定せねばならないという問題点があった。また、予めユーザより指定された実行要求タスク、即ちタイムスロット管理テーブルに割付けられたタスクしか実行できないので、周期的タスクとともに、エラー処理等の非周期的に発生する割込み処理を行なうことができないという問題点があった。

【0010】この発明は上記のような課題を解決するためになされたものであり、従来の周期的スケジューリングに加え、実行周期が最小基準周期未満のタスクが存在する場合のスケジューリングを提供し、全ての周期のタスクのスケジューリングを可能にした計算機システムを得ることを目的とする。また、周期的タスクとともに、エラー処理等の非周期的に発生する割込み処理も可能に

した計算機システムを得ることを目的とする。

【0011】

【課題を解決するための手段】この発明の第1の発明に係る計算機システムは、固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てするタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなかを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てする周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てするプライオリティスケジューリングモードとを切替えるスケジューリングモード切替手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切替手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切替え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てするようにしたものである。

【0012】また、第2の発明に係る計算機システムは、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てするタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理の優先度、即ちプライオリティに基づき割込み処理が周期的タスクよりプライオリティが高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させるようにしたものである。

【0013】一方、第3の発明に係る計算機システムは、上記第1の発明同様、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てするタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなかを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てする周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てするプライオリティスケジューリングモードとを切替えるスケジューリングモード切替手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切替手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切替え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てするようにしたものである。

【0014】

【作用】第1の発明における計算機システムは、タイムスロット割当て手段が実行レベルのタイムスロット割当てを実行するに先立って、上記実行周期判別手段、分類手段及び比較手段により、最小基準周期未満の実行周期のタスクの存在をチェックする。もし存在した場合は、スケジューリングモード切替手段が通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切替える。タイムスロット割当て手段は、スケジューリングモードがプライオリティスケジューリングモードとなっていることを認識すると、全てのタイムスロットに最小実行周期の実行レベルを割当てする。スケジューリング実行手段は、全てのタイムスロットで、割当てられた最小実行周期の実行レベルから順にタスクを実行する。全てのタイムスロットに最小実行周期の実行レベルを割当てすることで、常に最小実行周期の実行レベル

から順にタスクをチェックすることになり、結果として、最小実行周期の実行レベルに最大のプライオリティを与え、タイムスロット単位でタスクの起動をチェックするプライオリティスケジューリングが実現される。

【0015】また、第2の発明における計算機システムは、スケジューリング実行手段が割込み発生手段からの割込みの有無をチェックし、割込みが無い場合は通常の周期的スケジューリングを行なう。割込みがあれば、周期的タスクと割込み処理のプライオリティに基づき、割込み処理のプライオリティが高ければ割込み処理手段に割込み処理を行なわせる。割込み処理のプライオリティが低ければ通常の周期的スケジューリングを行なう。これにより、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御が可能となる。

【0016】一方、第3の発明における計算機システムは、ハードウェアタイマからのクロックにより割込み発生器及びカウンタが動作し、割込み発生器からスケジューリング実行手段へタイムスロット時間毎に割込みが発生するとともに、カウンタからレジスタに各時刻のタイムスロット値がセットされる。スケジューリング実行手段は、タイムスロット時間毎の割込みが入るとレジスタを見に行き、現時刻に対応するタイムスロット値を得る。このタイムスロット値を用いて、上記第1の発明と同様のスケジューリング処理を行なう。タイマにハードウェアタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。

【0017】

【実施例】実施例1. 以下、この発明の実施例を図について説明する。図1は、実施例1による計算機システムのスケジューリングに関する部分の構成図である。図において、26は新たに設けられたスケジューリングモード選択部であり、タスク情報12から、起動タスクの実行周期中、最小基準周期未満のタスクの有無をチェックし、もし存在すればスケジューリングモードとしてプライオリティスケジューリングモードを選択し、存在しなければ従来の周期的スケジューリングモードを選択する。27は上記スケジューリングモード選択部26が選択結果として出力するスケジューリングモード信号である。11は、上記スケジューリングモード選択部26からスケジューリングモード信号17を受け取り、周期的スケジューリングモードなら従来のものと同様の動作を行い、プライオリティスケジューリングモードなら、全てのタイムスロットに実行レベル1を割当てるタイムスロット割当て部であり、この機能は図17に示したタイムスロット割当て手段11cにより実現される。その他については従来のシステムと同様である。

【0018】図2は上記スケジューリングモード選択部26の内部構成を示す機能ブロック図である。図におい

て、26aはタスク情報12に基づき各タスクの実行周期を判別する実行周期判別手段、26bは上記実行周期判別手段26aの判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段、26cは上記分類手段26bにより最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなかを判定する比較手段、26dはこの比較結果に基づき、従来からの周期的スケジューリングモードと、新たなスケジューリングモードとして全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切替えるスケジューリングモード切替手段であり、このスケジューリングモード切替手段26dの出力がスケジューリングモード信号27となる。なお、上記実行周期判別手段26aと分類手段26bは、従来からタイムスロット割当て部11に備わっている実行周期判別手段11a及び分類手段11b(図17参照)と同様なものである。従って、これらを一組だけ備えてタイムスロット割当て部11とスケジューリングモード選択部26で共用することも可能である。

【0019】次に、本実施例におけるタイムスロット割当て方法について図3、図4のフロー図を用いて説明する。まず、スケジューリングモード選択部26は、実行周期判別手段26aにより全ての起動タスクの実行周期を判別し、分類手段26bにより実行周期別に実行レベルに分類する(ST31)。例えば、実行周期50msのタスクは実行レベル1に、実行周期70msのタスクは実行レベル2に、という具合に分類する。次に、スケジューリングモード選択部26は、比較手段26cにより最小実行周期のタスクが所属する実行レベル1の実行周期と最小基準周期を比較し(ST32)、実行レベル1の周期の方が小さければ、スケジューリングモード切替手段26dによりプライオリティスケジューリングモードを選択し(ST34)、大きければ従来同様の周期的スケジューリングモードを選択する。(ST33)。次に、タイムスロット割当て部11は、スケジューリングモード信号27を受け取り、このモードをチェックし(ST35)、これがプライオリティスケジューリングモードの場合は、図5に示すように、タイムスロット管理テーブル211の全てのタイムスロット1~Nに実行レベル1を割当てる(ST36)。スケジューリングモードが周期的スケジューリングモードの場合は、従来のシステムと同様の処理を行なう。即ち、タイムスロット割当て部11は、実行レベル1~Mについて以下の処理を行う(ST37)。実行レベル1について、実行レベル1の全てのタスク実行時間の総和に、実行レベル1の実行周期の最小基準周期に対する割合を掛けることで、実行レベル1のタスクの実行時間合計が最小基準周期内に占める時間を求める(ST38)。従来例でも述べたように、例えば、実行レベル1のタスクの実行周期が1

00ms、実行レベル1のタスクの実行時間合計が40ms、最小基準周期が50msの時、 $40 \times (50/100) = 20\text{ms}$ が求める値となる。即ち、実行レベル1は最小基準周期内で20msの実行時間が必要となる。その結果をタイムスロット時間で割り、小数点以下を切上げ、最小基準周期内で実行レベル1に必要なタイムスロット数を求める(ST39)。前述の例では、タイムスロット時間が10msの時、 $20/10 = 2$ 、故に2個のタイムスロットが必要となる。以上の処理を同様に実行レベル2からMまで繰り返し(ST40)、各実行レベルに必要なタイムスロット数を求める。次にタスクの実行に必要なシステムの実行時間に必要なタイムスロット数を求める(ST41)。以上で、各々の実行レベルが必要とするタイムスロット数が算出できるが、ただし、その合計が最小基準周期のタイムスロット数Nを越えると、実行最大負荷を越え制御不可となるため、その合計とNを比較し(ST42)、合計がタイムスロット数Nを越えている場合は、実行レベル1から順に必要なタイムスロット数だけタイムスロット管理テーブル211に格納し、テーブルが満たされたとき、残りのレベルは無視するようにする(ST44)。Nのほうが大きい場合は問題なく、実行レベル1から順にタイムスロット管理テーブル211に格納する(ST43)。以上でタイムスロットへの実行レベルの割当てを終了する。

【0020】タイムスロット割当て後のスケジューリング実行は、従来の手段と同様の動作を行う。この時、プライオリティスケジューリングモードで図5に示したようにタイムスロット管理テーブル211が生成された場合は、従来例で示した図21のフロー図は図6に示すようになり、全てのタイムスロットで実行レベル1を得て(ST212)、実行レベル1から図22に従ってレベル管理テーブル212a~212cをチェックし、最初に発見したキューイングタスクを実行する。この結果、実行レベル1に最大のプライオリティを与え、実行レベルMに最小のプライオリティを与えることになる。即ち、どのタイムスロットにおいても実行レベル1を獲得し、図22の実行レベル探索に従い実行タスクを決定するので、実行レベルの小さいタスクが終了しない限り、大きいタスクは実行されない。この結果、実行レベルの小さいタスクほど高いプライオリティを与えるプライオリティスケジューリングが実現される。この場合、従来例で示した図16は図7に示すようになる。

【0021】実施例2. 図8は実施例2の計算機システムのスケジューリングに関する部分の構成図である。本実施例は、図1の実施例1のものに、エラー発生等の非周期的イベントの発生をスケジューリング実行部13に割込みで知らせる割込み発生部28と、割込みが生じた時の処理を行なう割込み処理部29と、予め定められた周期的タスクと上記割込み処理との間のプライオリティ

情報を格納しておくプライオリティ情報格納部30とを設けたものである。そして、スケジューリング実行部13は、通常は実施例1のような周期的スケジューリングを行い、エラー発生等により割込み発生部28より割込み(割込み種類を含む)31が入った時は、プライオリティ情報格納部30からのプライオリティ情報32に基づき、その割込み処理が周期的タスクよりプライオリティが高ければ、割込み処理要求(割込み種類を含む)33を割込み処理部29へ送り、周期的タスクの実行を一時中断して割込み処理を行なうようにしたものである。

【0022】上記割込み発生部28は、図9に示すように、割込み種類通知テーブル28aと、割込み発生通知テーブル28bと、実行タスク監視部28cとを有しており、以下、エラー割込みを例にとって説明する。計算機システムの実行中タスク28dは自タスク内に何らかのエラーが生じると、その割込み種類28eを割込み種類テーブル28aに書き込み、割込み発生通知テーブル28bには割込み発生信号28fをセットする。一方、実行タスク監視部28cは、割込み発生通知テーブル28bを一定周期でポーリングし、該テーブル28bからの割込み発生信号28gがセットされると、割込み種類通知テーブル28aから割込み種類28hを獲得し、それを割込み31によりスケジューリング実行部13へ送る。処理終了後は、再び割込み発生通知テーブル28bのポーリングに入る。一方、上記割込み処理部29は、図10に示すように、割込み種類通知テーブル29aと、割込み処理要求通知テーブル29bと、割込みハンドラ29cと、割込み種類に対応した複数の割込み処理ルーチン(1)29d~(n)29fとを有している。スケジューリング実行部13は、割込み発生部28から割込み種類を含む割込み31を受けると、割込み種類に対応する割込み処理のプライオリティと周期的タスクのプライオリティを比較し、割込み処理のプライオリティの方が高ければ、割込み処理部29の割込み種類通知テーブル29aに該割込み種類33bを書き込み、その後、割込み処理要求通知テーブル29bに割込み処理要求33aをセットする。一方、割込みハンドラ29cは、割込み処理要求通知テーブル29bを一定周期でポーリングし、もし該テーブル29bからの割込み処理要求29gがセットされると、割込み種類通知テーブル29aから割込み種類29hを獲得し、それに応じて割込み処理起動信号29i~29kを出力し割込み処理ルーチン(1)29d~(n)29fのうち1つを起動する。処理が終了すると、再び割込み処理要求通知テーブル29bのポーリングに入る。

【0023】次に、本実施例のスケジューリング動作について、図11のフロー図を用いて説明する。スケジューリング実行部13が割込みの有無をチェックし(ST111)、割込みが無い場合は実施例1同様の周期的スケジューリングを行なう。割込みがあれば以下の

処理を行なう。割込み処理と周期的タスクの間のプライオリティが予め格納されたプライオリティ情報格納部30から、割込み処理と周期的タスクのプライオリティ情報32を得てこれらと比較し(ST112)、割込み処理のプライオリティが高ければ、割込み処理を実行し

(ST114)、割込みをクリアする(ST115)。割込み処理のプライオリティが低ければ、割込みをペンディングし(ST113)、実施例1における通常の周期的スケジューリングを行なう。ただし、この周期的スケジューリングにおいて、実行タスクが無い場合(ST124)、このタイムスロットを利用してペンディング割込みの処理を行なう(ST125→ST126)。割込み処理後は割込みをクリアする(ST127)。以上の動作により、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御も可能となる。即ち、従来例や実施例1においては、予めユーザにより指定されてタイムスロット管理テーブル211に割付けられたタスクしか実行できないが、上記のような割込み発生部28と割込み処理部29を付加することで、非周期的に発生するエラー処理などを、そのプライオリティを予め高くしておくことにより、周期的スケジューリングを一時中断して実行することができる。なお、本実施例では、図1の実施例1のものに上記のような割込み処理機能を付加したものについて説明したが、図15の従来例のものに適用することも可能である。

【0024】実施例3。図13は実施例3の計算機システムのスケジューリングに関する部分の構成図である。本実施例は、図1の実施例1におけるS/Wタイマ14、タイムスロット計算部16及びタイムスロット情報格納部18をハードウェアで構成したものである。即ち、34は図1のS/Wタイマ14の代りとなるハードウェア(以下、H/Wと略記する)タイマで、計算機システムが元々有するものである。35は上記H/Wタイマ34からのクロック36に基づきタイムスロット時間毎にスケジューリング実行部13に割込み37を発生する割込み発生器、38~40は上記H/Wタイマ34からのクロック36に基づき各タイムスロット値1~NをそれぞれカウントするN個のカウンタで、これら割込み発生器35及びカウンタ38~40は図1のタイムスロット計算部16の代りに設けられたものである。41は上記各カウンタ38~40の出力42により各タイムスロット値1~Nに対応するビットがセットされ、上記割込み発生器35からの割込み発生毎にスケジューリング実行部13により現時刻のタイムスロット値43が読み出されるタイムスロットレジスタで、これは図1のタイムスロット情報格納部18の代りに設けられたものである。

【0025】次に、本実施例のスケジューリング動作について、図14のフロー図を用いて説明する。H/Wタ

イマ34のクロック36を用い、割込み発生器35はタイムスロット時間毎にスケジューリング実行部13に割込み37を発生する。カウンタ(1)38からカウンタ(N)40はH/Wタイマ34のクロック36を利用して、それぞれ定められたタイムスロット値1~Nをカウントし、最小基準周期毎にタイムスロットレジスタ41の対応ビットをセットする。スケジューリング実行部13は、割込み発生器35からタイムスロット時間毎の割込み37が入るとタイムスロットレジスタ41を見に行き、セットされたビットからタイムスロット値43を得る(ST141)。このタイムスロット値を用いて、実施例1と同様のスケジューリング処理を行ない、H/Wタイマ割込みを待つ(ST145)、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。タイマにH/Wタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。即ち、実施例1では、オペレーティングシステムのS/Wタイマ機能を用い、現在の時刻を獲得する処理、タイムスロットの計算、スケジューリング実行部13へのタイムスロット変更割込みの通知といった処理を全てソフトウェアで実現している。この場合、タイムスロットの前回値の保持が必要な上、オーバーヘッド時間(図7のシステム時間)が大きくなるという欠点がある。これに対し、本実施例のように、以上の機能を、H/Wタイマを用いて割込み発生器、カウンタ及びレジスタなどのハードウェアで実現した場合、タイムスロットの前回値保持が不要となる上、オーバーヘッドを低減でき、高速なスケジューリングが可能となる。

【0026】

【発明の効果】以上のように、この発明の第1の発明によれば、固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかなどを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切り換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段

とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたので、可能な限り周期的スケジューリングを行うことでプロセッサへの負担を均等にするとともに、スケジューリング不可能な周期のタスクが存在する場合は、それを最優先で実行することになり、全ての周期のタスクに対応したスケジューリングが可能になるという効果が得られる。

【0027】また、第2の発明によれば、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理のプライオリティに基づき割込み処理が周期的タスクよりプライオリティが高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させるようにしたので、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御が可能となる効果が得られる。

【0028】一方、第3の発明によれば、上記第1の発明同様に、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いかな否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切替えるスケジューリングモード切換手段と、現時刻に対応するタイムス

ロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたので、上記第1の発明と同様な効果が得られるとともに、タイマにH/Wタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる効果が得られる。

【図面の簡単な説明】

【図1】この発明の実施例1による計算機システムのスケジューリングに関する部分の構成図である。

【図2】図1のスケジューリングモード選択部の内部構成を示す機能ブロック図である。

【図3】図1のスケジューリングモード選択部とタイムスロット割当て部の動作を示すフロー図である。

【図4】図3の続きを示すフロー図である。

【図5】プライオリティスケジューリング時のタイムスロット対応タスク情報テーブル内の詳細図である。

【図6】プライオリティスケジューリング時のスケジューリング実行部の動作を示すフロー図である。

【図7】プライオリティスケジューリング時のタスクの実行状態を示す図である。

【図8】この発明の実施例2による計算機システムのスケジューリングに関する部分の構成図である。

【図9】図8の割込み発生部の内部構成図である。

【図10】図8の割込み処理部の内部構成図である。

【図11】図8のスケジューリング動作を示すフロー図である。

【図12】図11の続きを示すフロー図である。

【図13】この発明の実施例3による計算機システムのスケジューリングに関する部分の構成図である。

【図14】図13のスケジューリング動作を示すフロー図である。

【図15】従来の計算機システムのスケジューリングに関する部分の構成図である。

【図16】スケジューリング時のタスクの実行状態を示

す図である。

【図17】タイムスロット割当て部の内部構成を示す機能ブロック図である。

【図18】タイムスロット対応タスク情報テーブル内の詳細図である。

【図19】図15のタイムスロット割当て部の動作を示すフロー図である。

【図20】図19の続きを示すフロー図である。

【図21】スケジューリング実行部の動作を示すフロー図である。

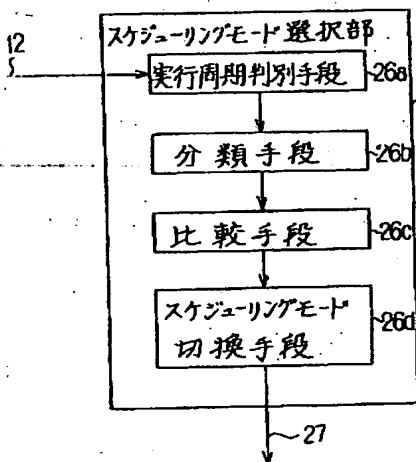
【図22】実行レベルのチェックの順番を示す図である。

【符号の説明】

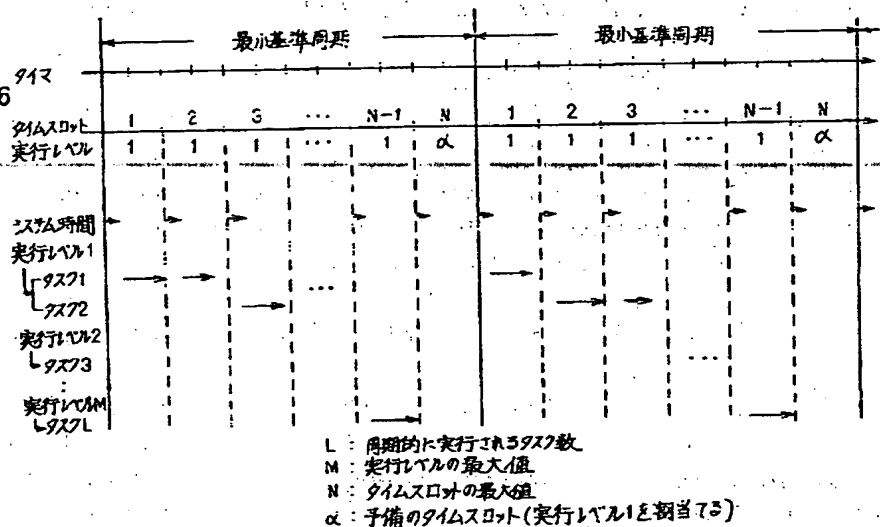
- 11 タイムスロット割当て部
- 11a 実行周期判別手段
- 11b 分類手段
- 11c タイムスロット割当て手段
- 12 タスク情報
- 13 スケジューリング実行部 (スケジューリング実行手段)
- 14 S/Wタイマ
- 15 定周期割込み
- 16 タイムスロット計算部
- 17 タイムスロット変更割込み
- 18 タイムスロット情報格納部
- 19 タイムスロット値 (変更値)
- 20 タイムスロット値 (現在値)
- 21 タイムスロット対応タスク情報テーブル

- * 22 テーブル情報 (現在値)
- 23 テーブル情報 (変更値)
- 24 起動タスクキューイング部
- 25 起動要求タスク
- 26 スケジューリングモード選択部
- 26a 実行周期判別手段
- 26b 分類手段
- 26c 比較手段
- 26d スケジューリングモード切換手段
- 10 27 スケジューリングモード信号
- 28 割込み発生部 (割込み発生手段)
- 29 割込み処理部 (割込み処理手段)
- 30 プライオリティ情報格納部
- 31 割込み (割込み種類を含む)
- 32 プライオリティ情報
- 33 割込み処理要求 (割込み種類を含む)
- 34 H/Wタイマ
- 35 割込み発生器
- 36 クロック
- 20 37 割込み
- 38~40 カウンタ
- 41 タイムスロットレジスタ
- 42 カウンタ出力
- 43 タイムスロット値
- 211 タイムスロット管理テーブル
- 212a~212c レベル管理テーブル
- 213a1~213an, 213b1, 213c1, 213c2 タスク管理テーブル

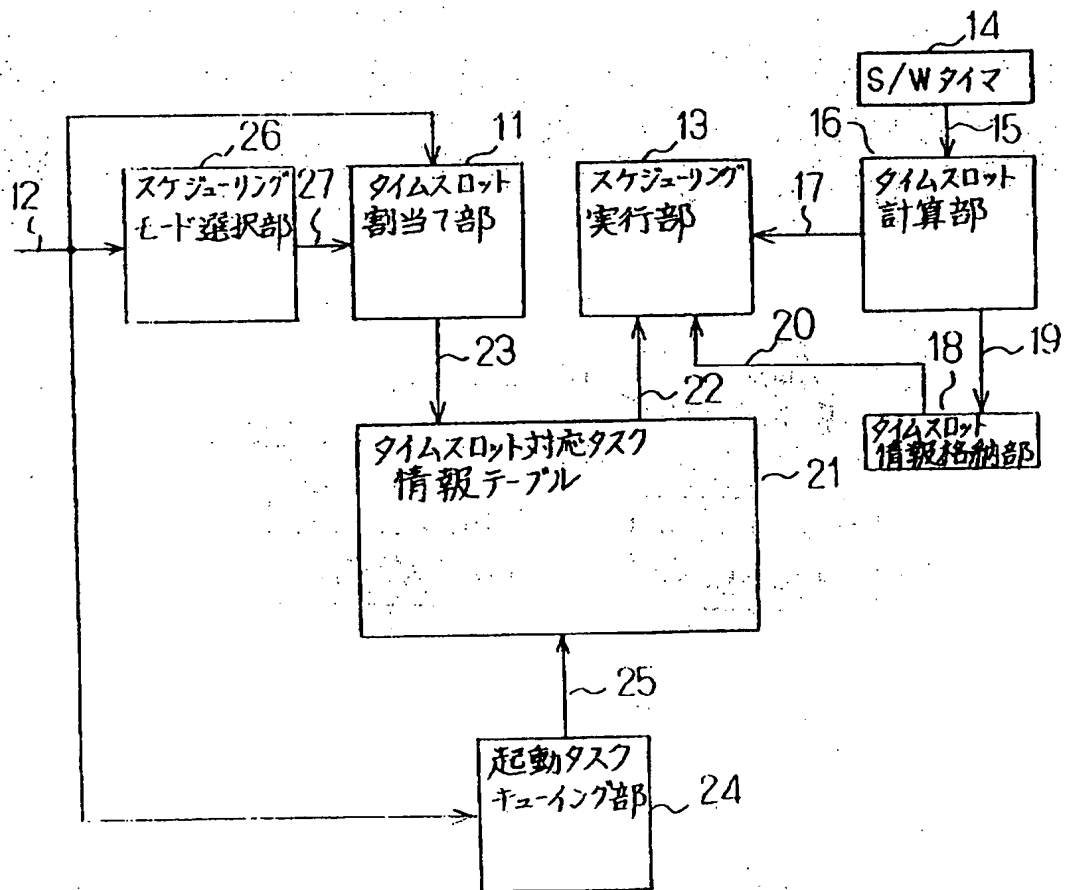
【図2】



【図7】

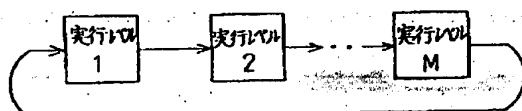


【図1】

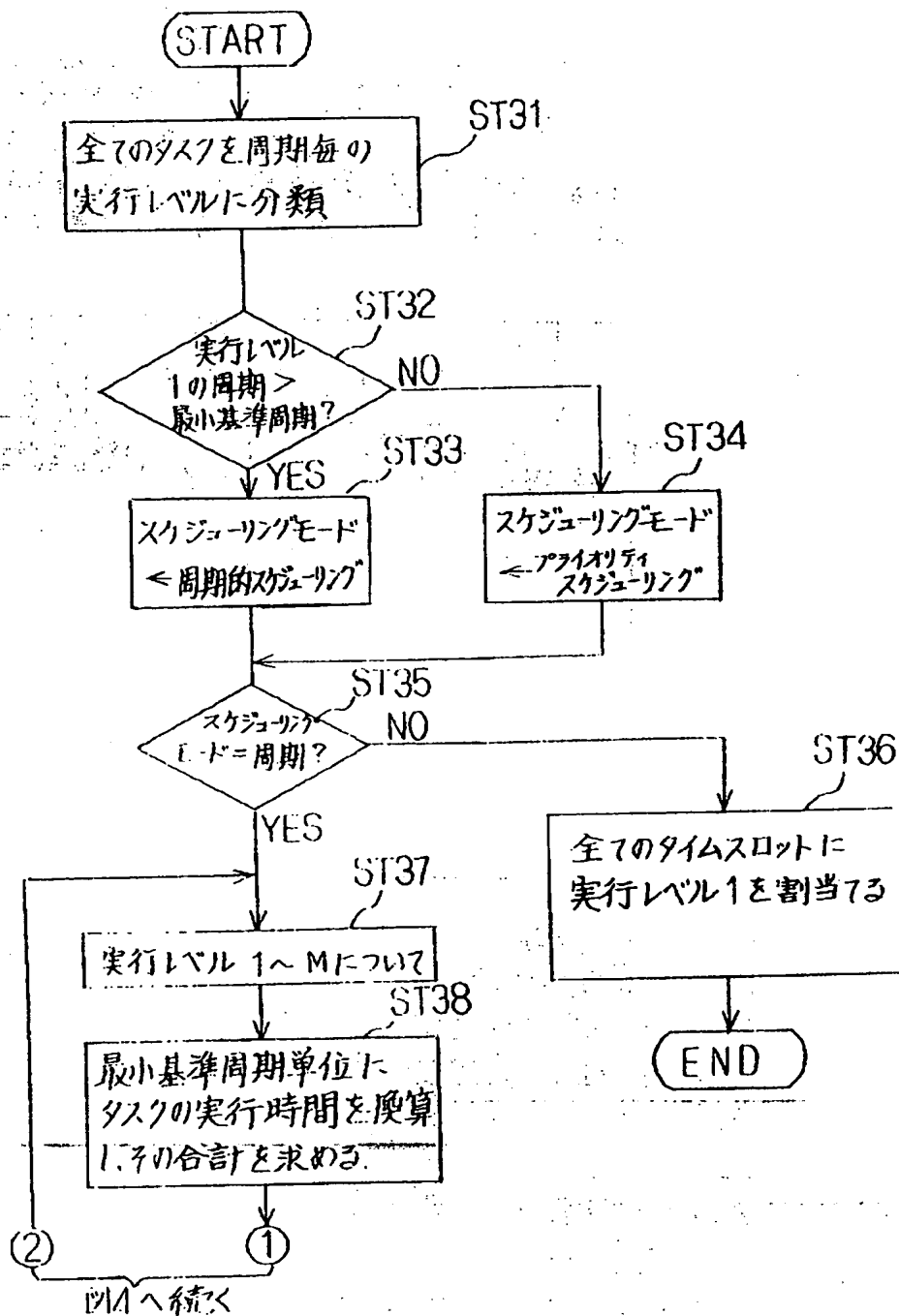


- 12 ; タスク情報
 15 ; 定周期割込み
 17 ; タイムスロット変更割込み
 19 ; タイムスロット値(変更値)
 20 ; タイムスロット値(現在値)
 22 ; テーブル情報(現在値)
 23 ; テーブル情報(変更値)
 25 ; 起動要求タスク
 27 ; スケジューリングモード信号

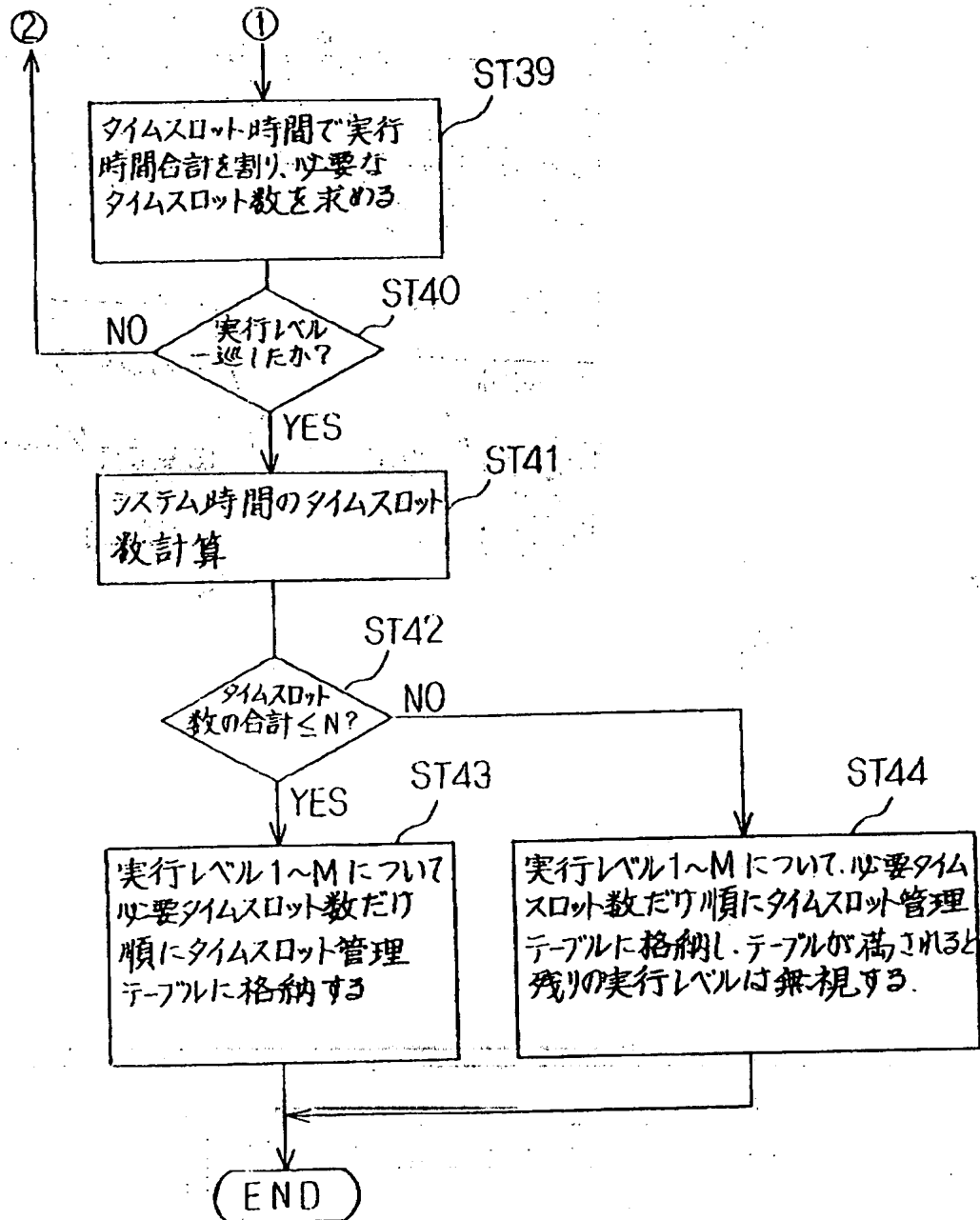
【図22】



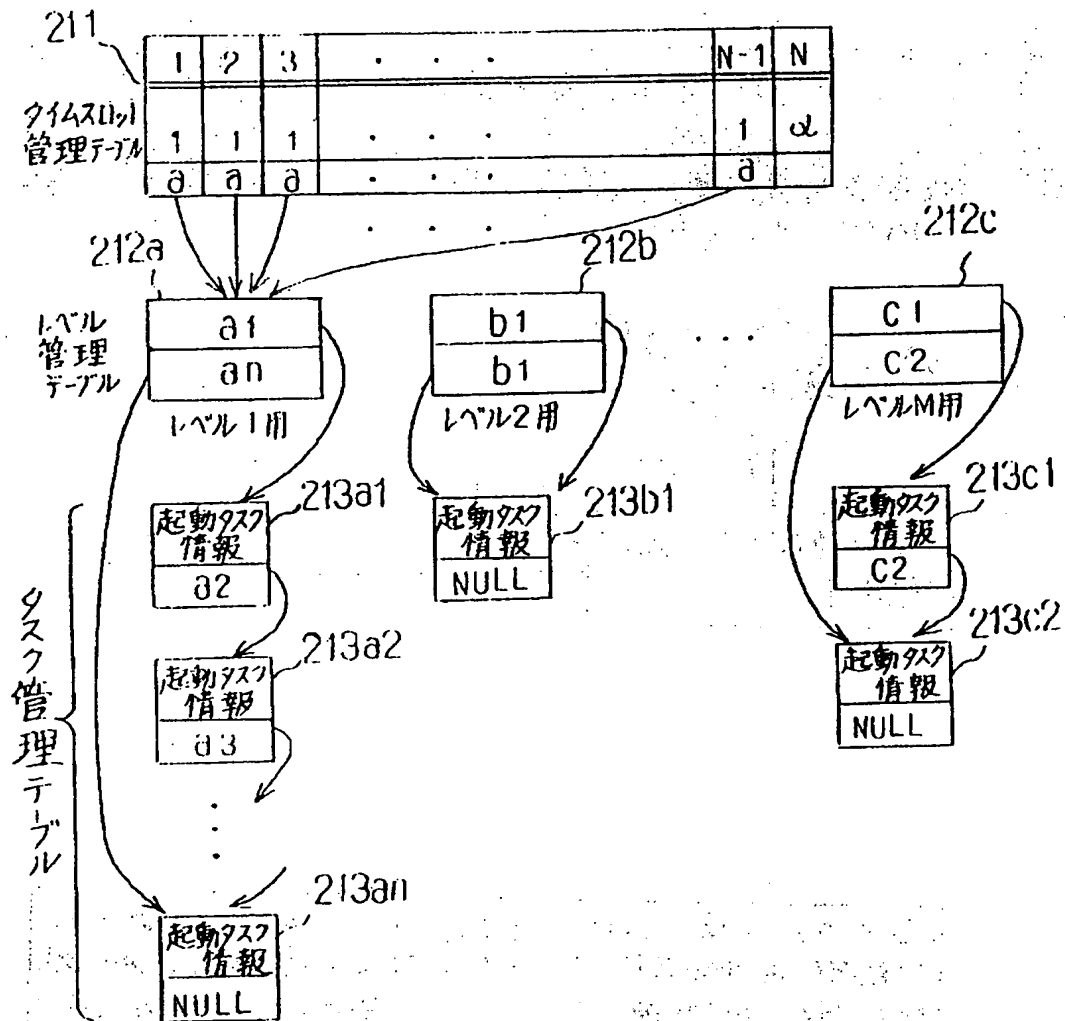
【図3】



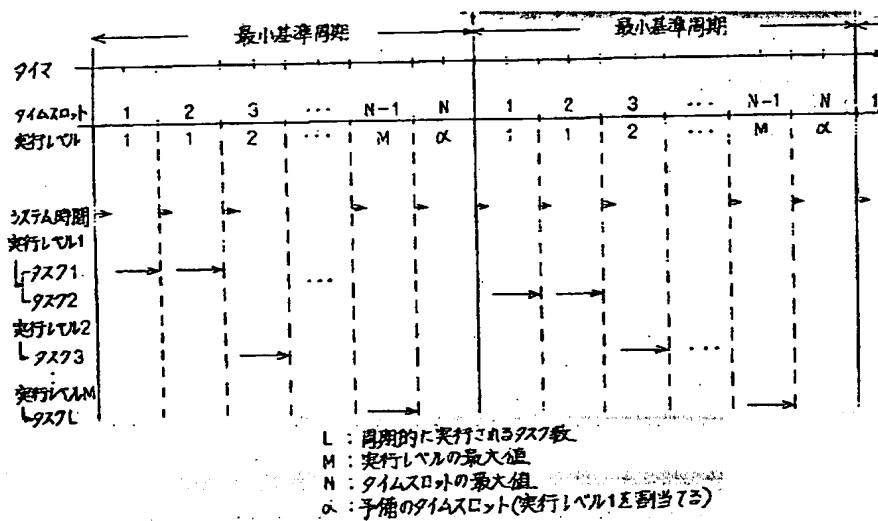
【図4】



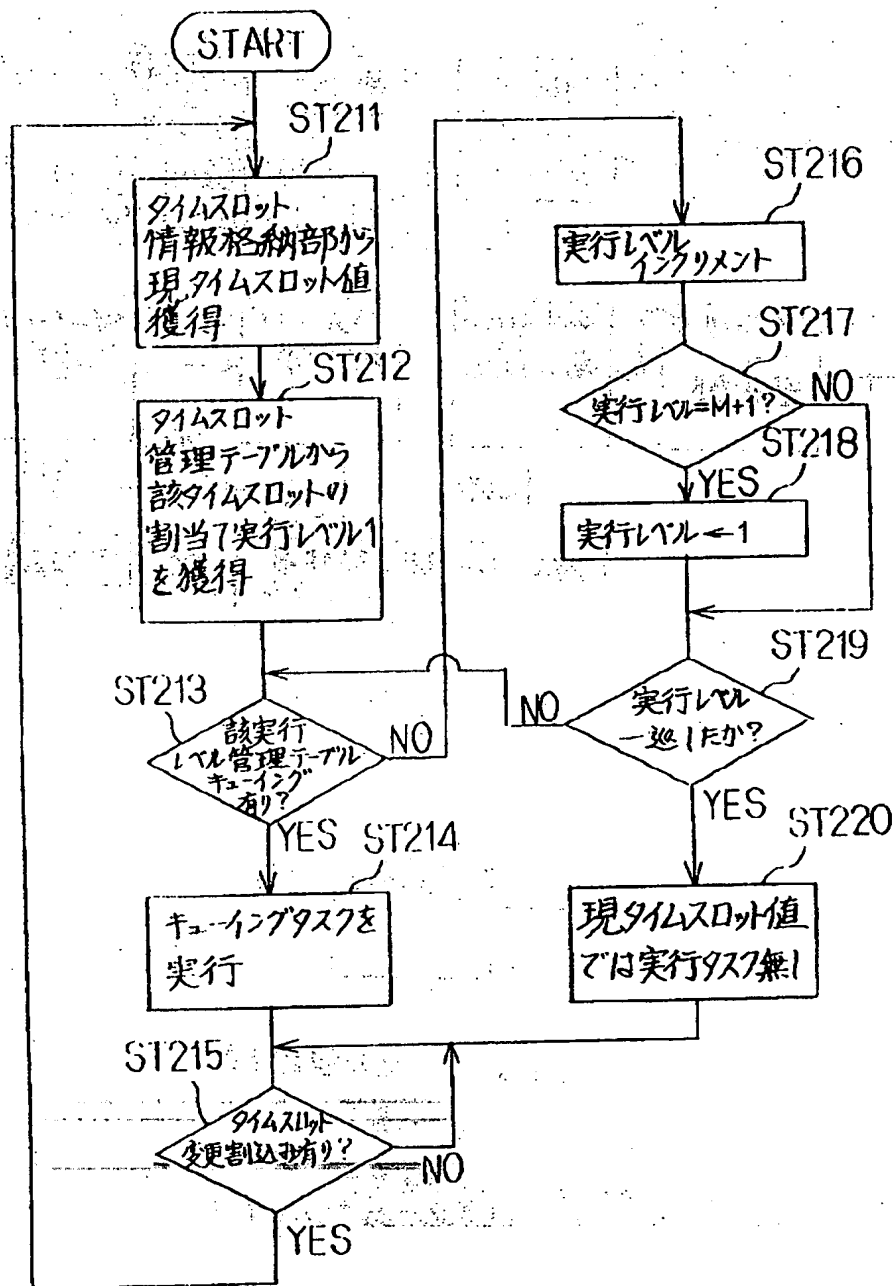
【図5】



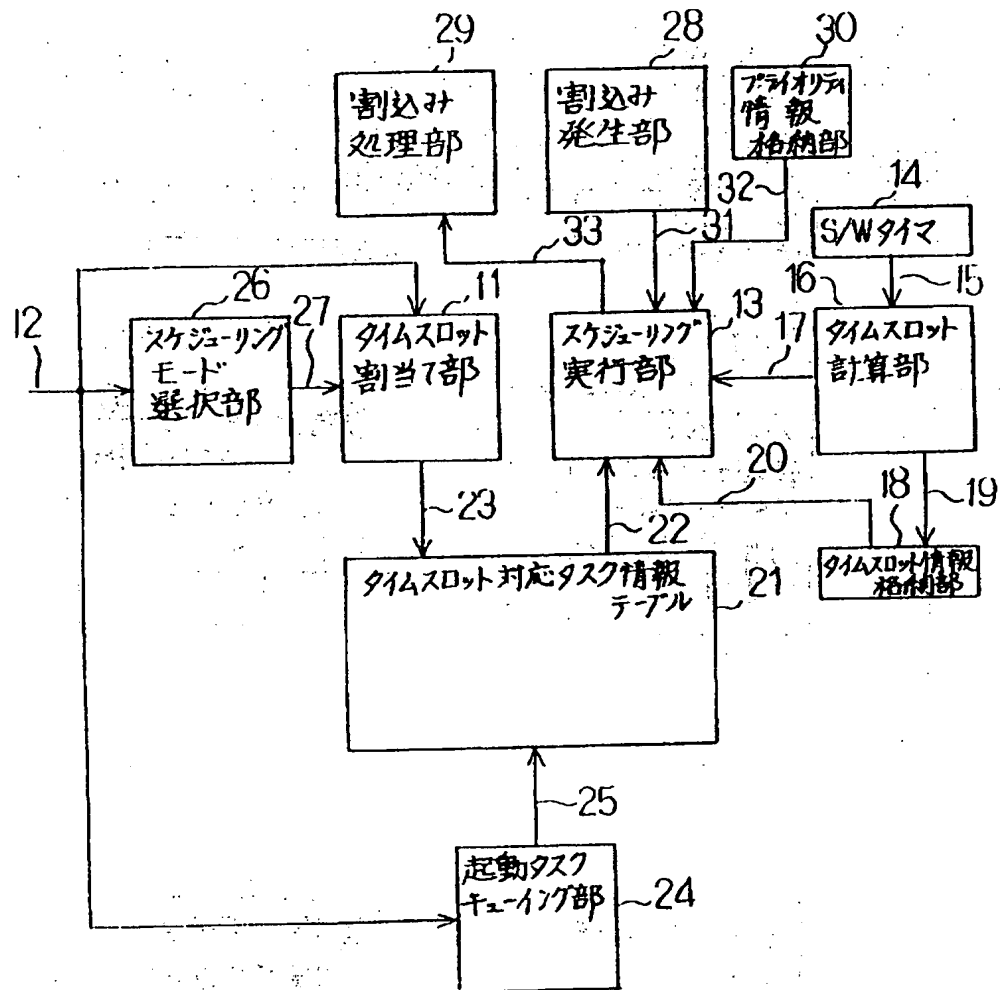
【図16】



【図6】



【図8】

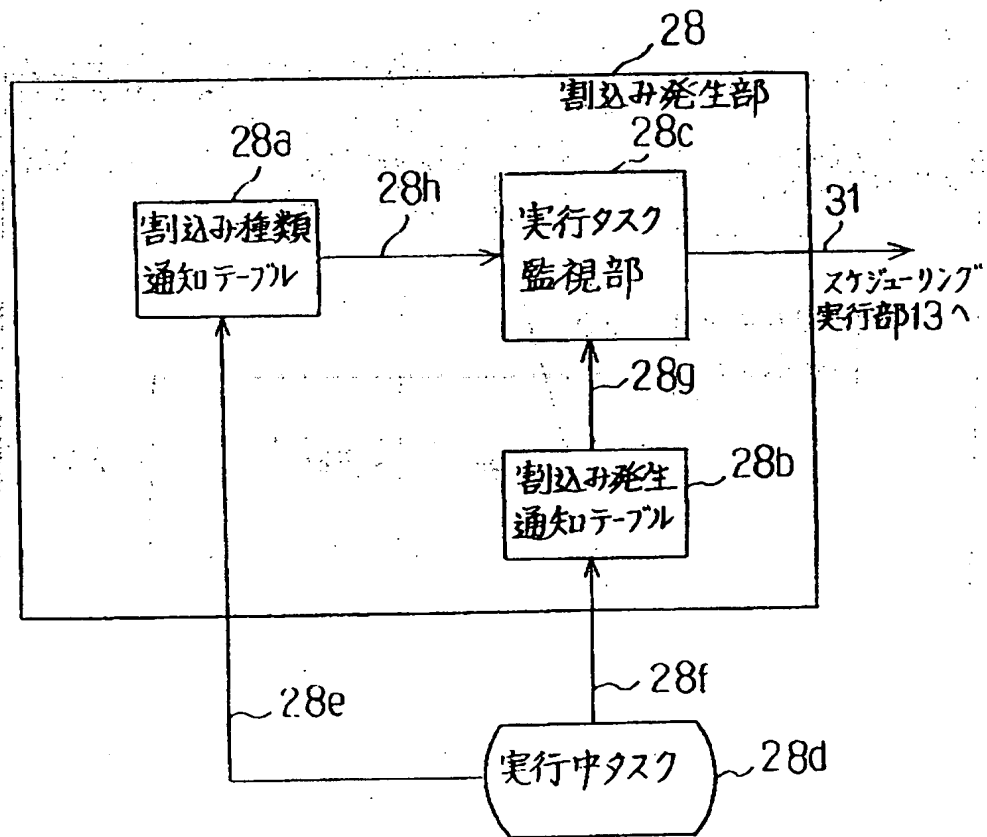


31; 割込み (割込み種類を含む)

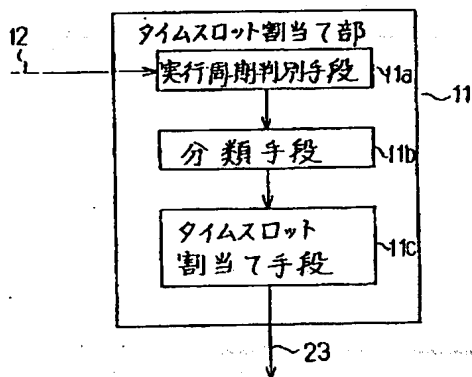
32; プライオリティ情報

33; 割込み処理要求
(割込み種類を含む)

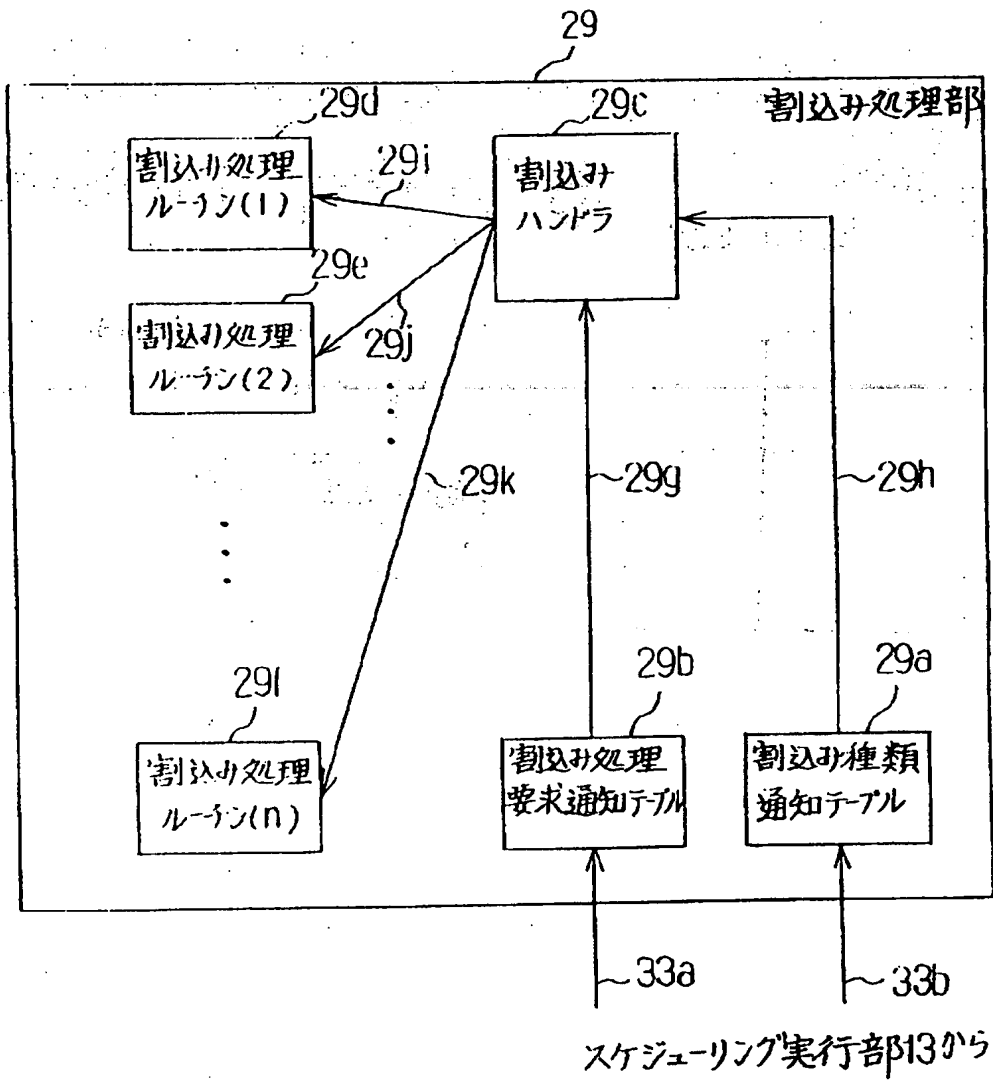
【図9】



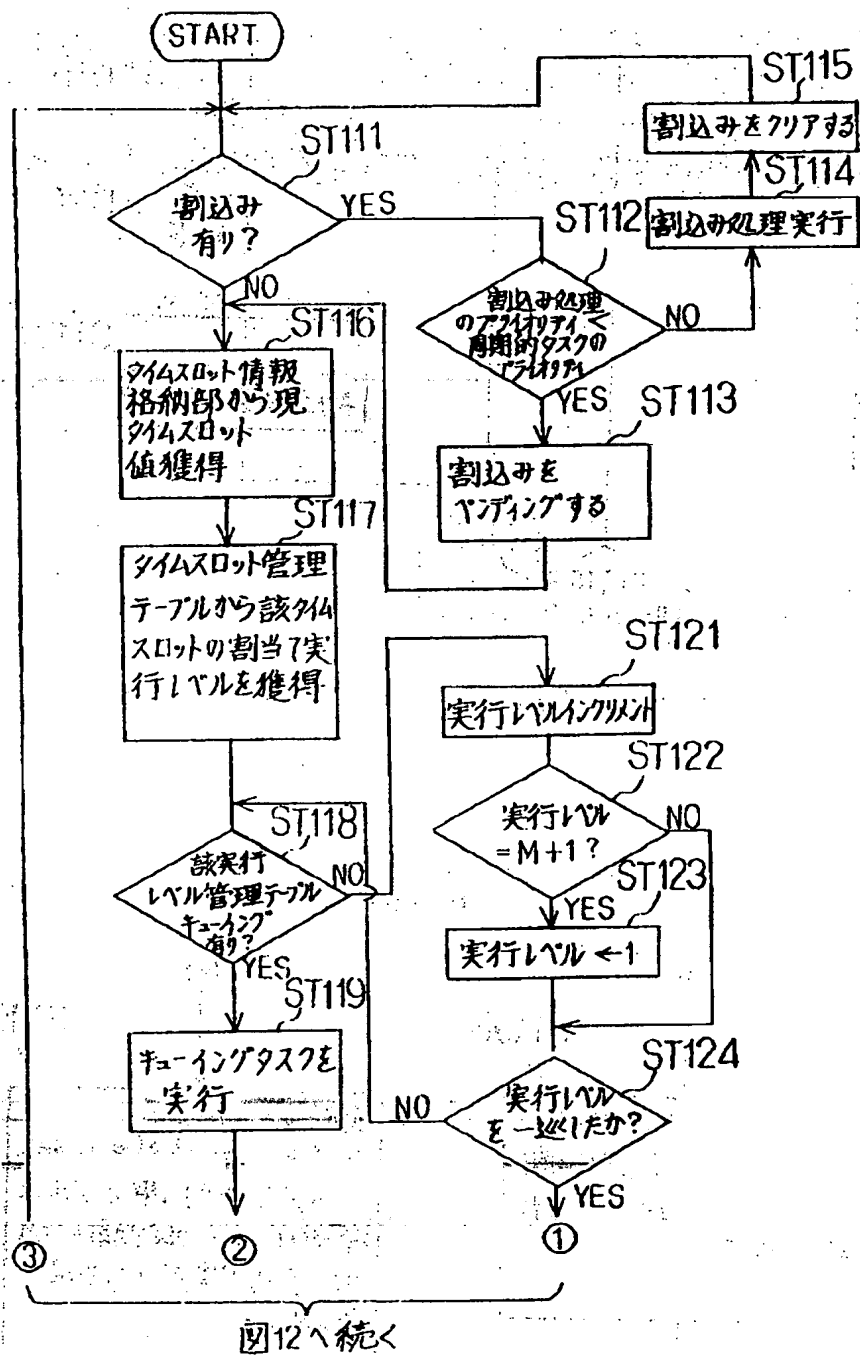
【図17】



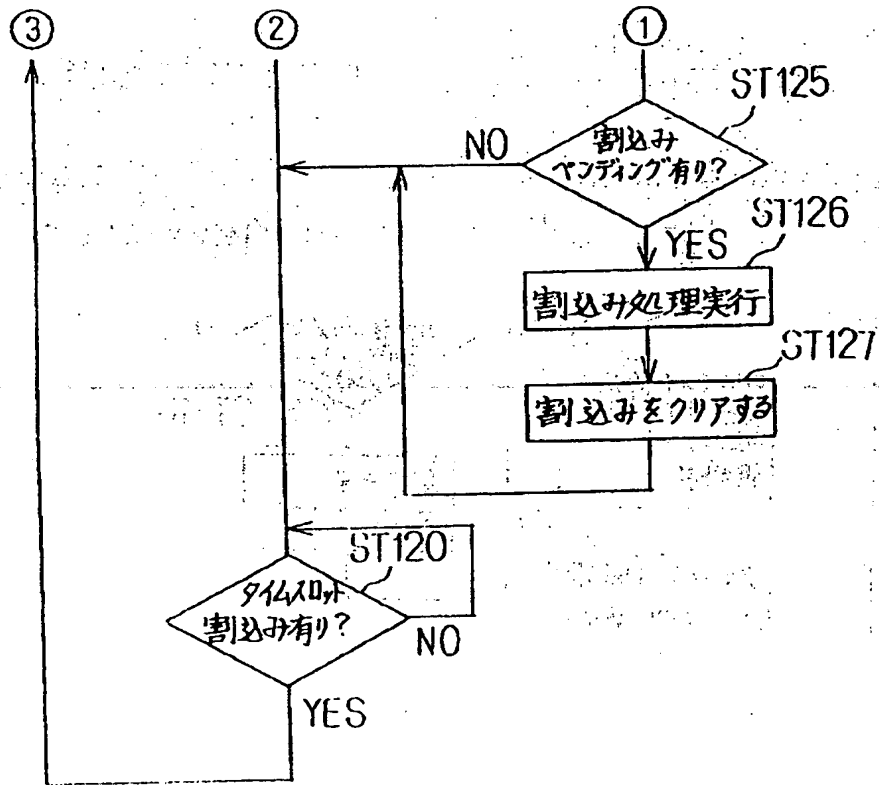
【図10】



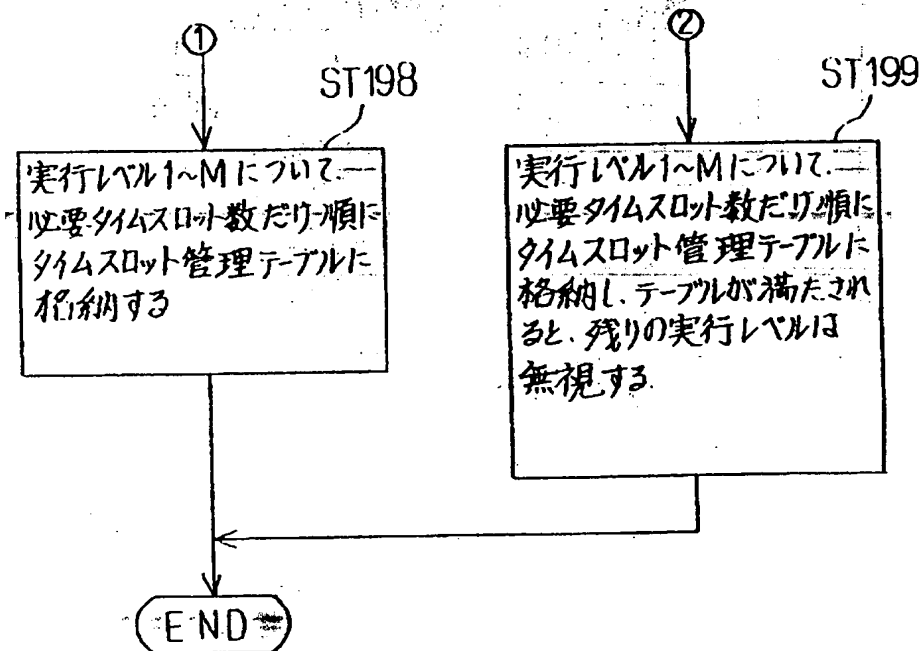
【図11】



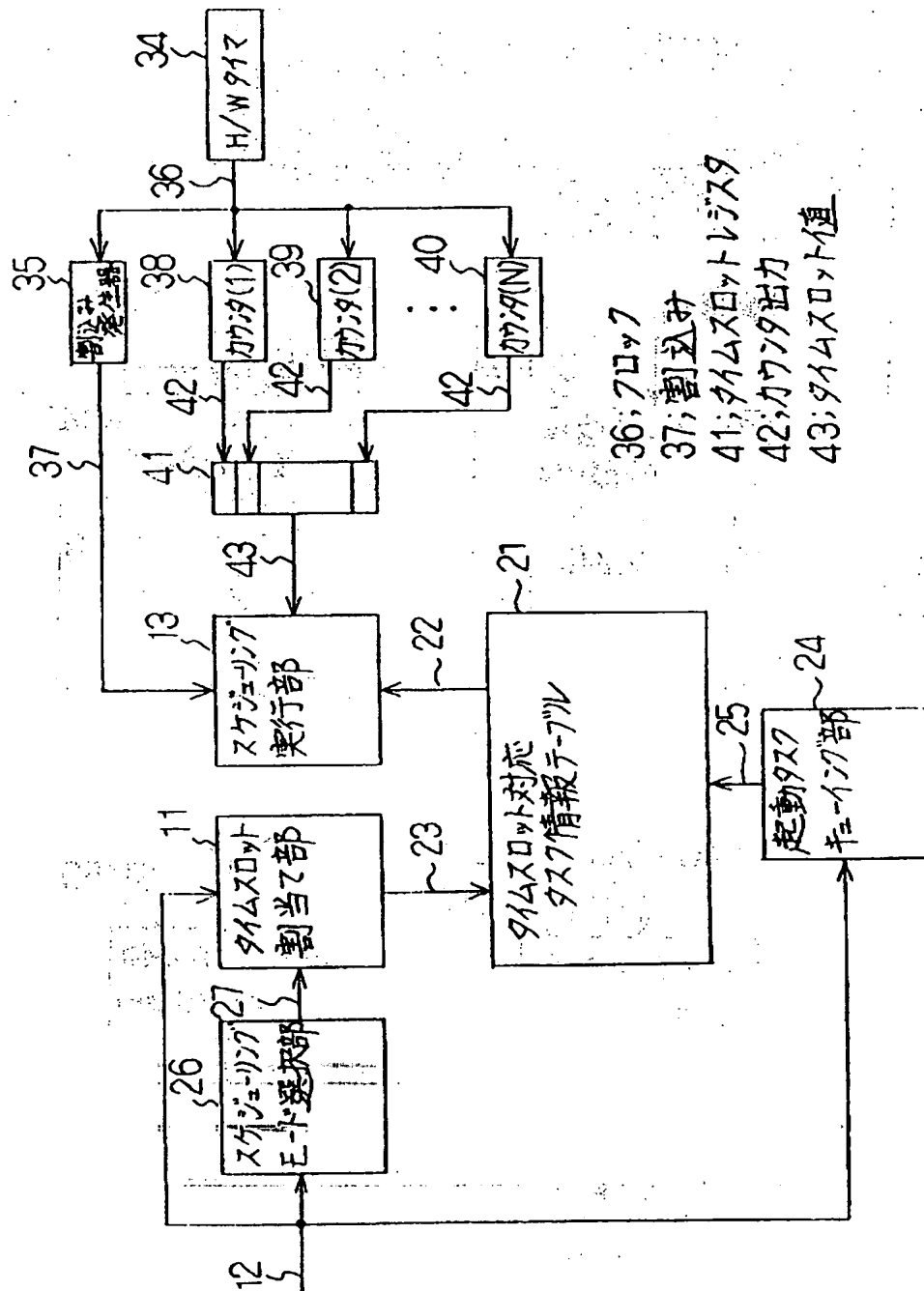
【図 12】



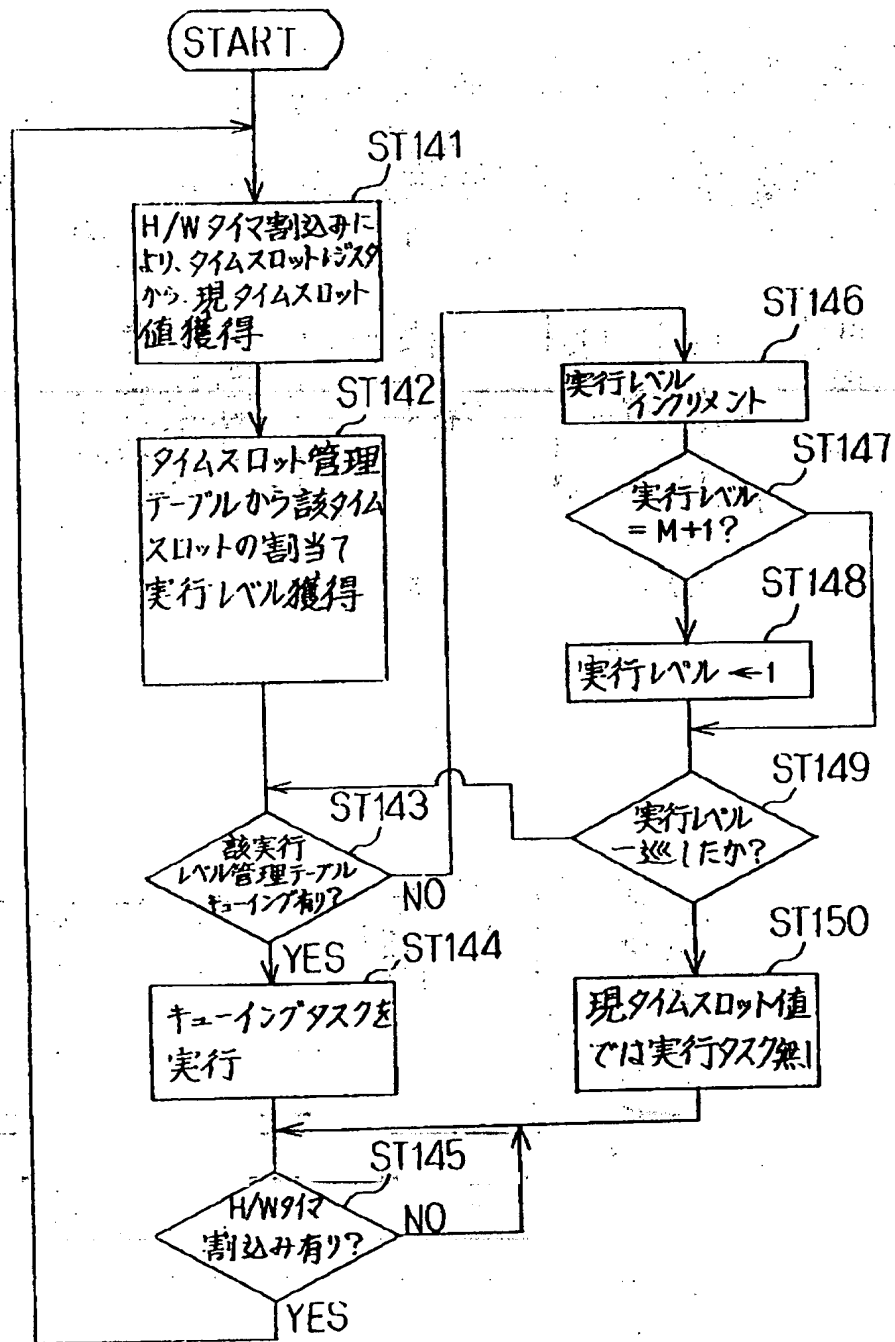
【図 20】



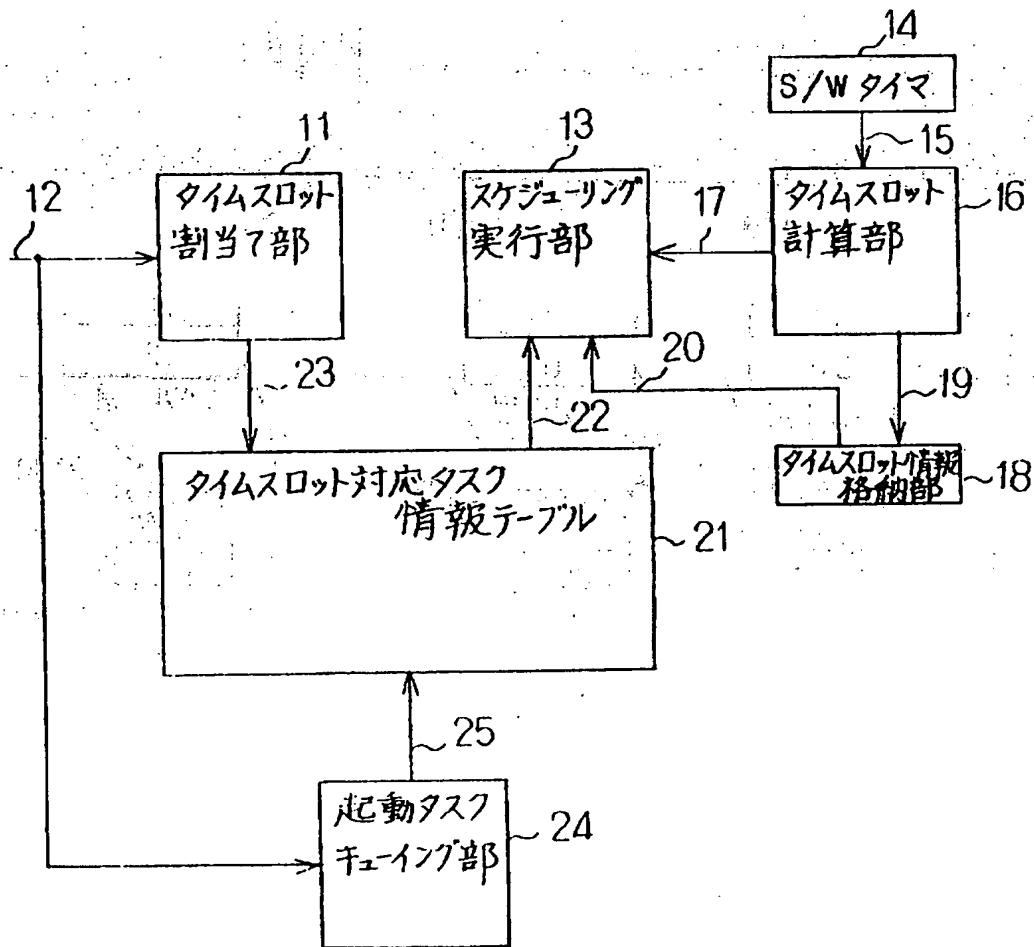
【図 13】



【図14】



【図15】



12; タスク情報

15; 定周期割込み

17; タイムスロット変更割込み

19; タイムスロット値-(変更値)

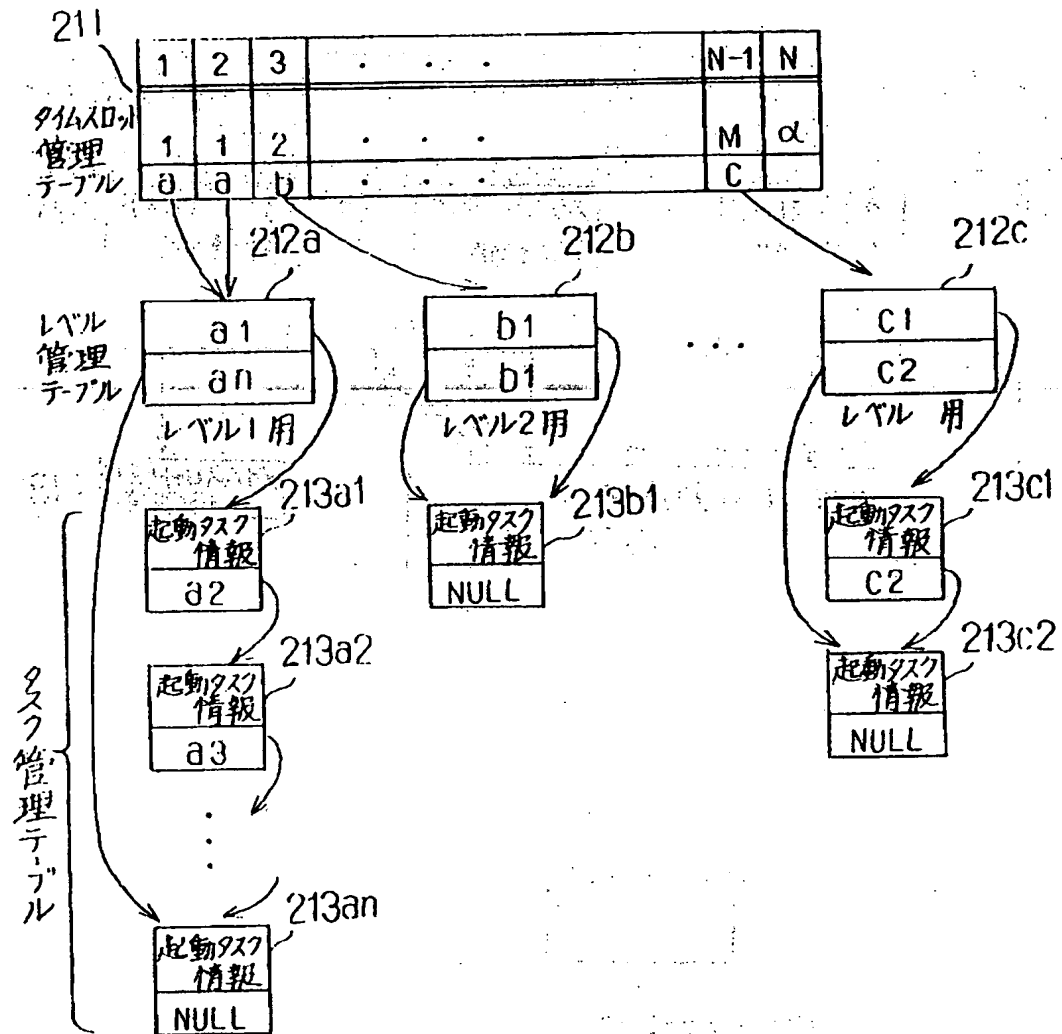
20; タイムスロット値(現在値)

22; テーブル情報(現在値)

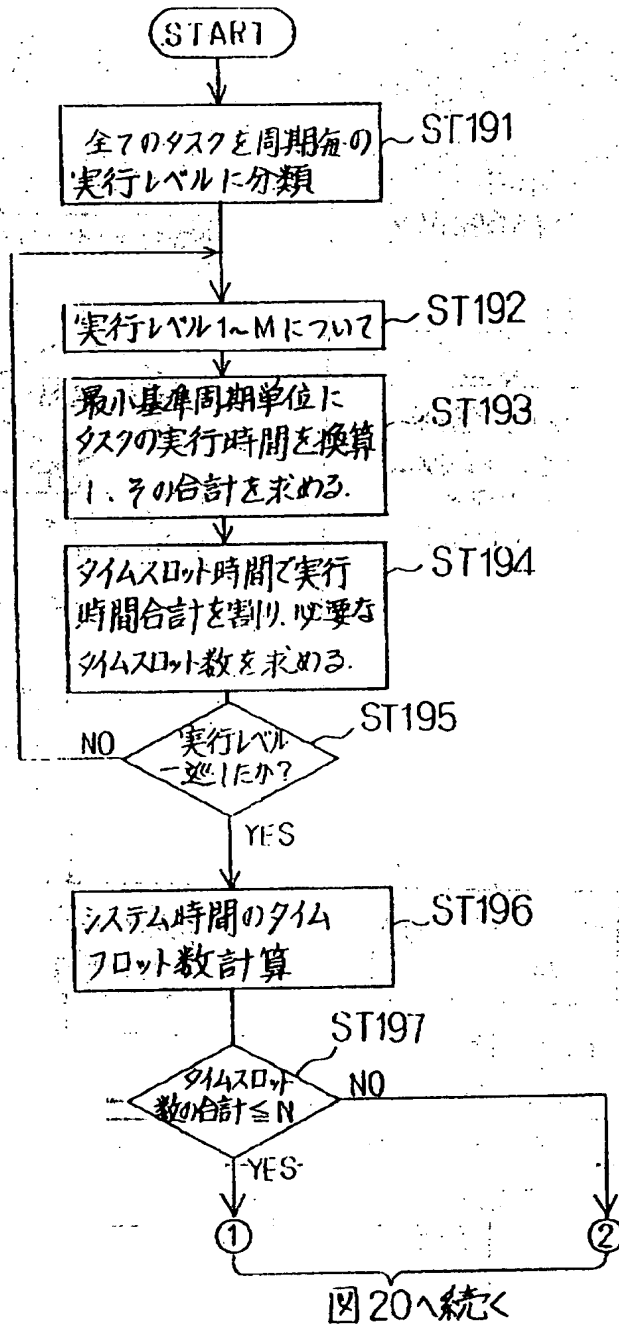
23; テーブル情報(変更値)

25; 起動要求タスク

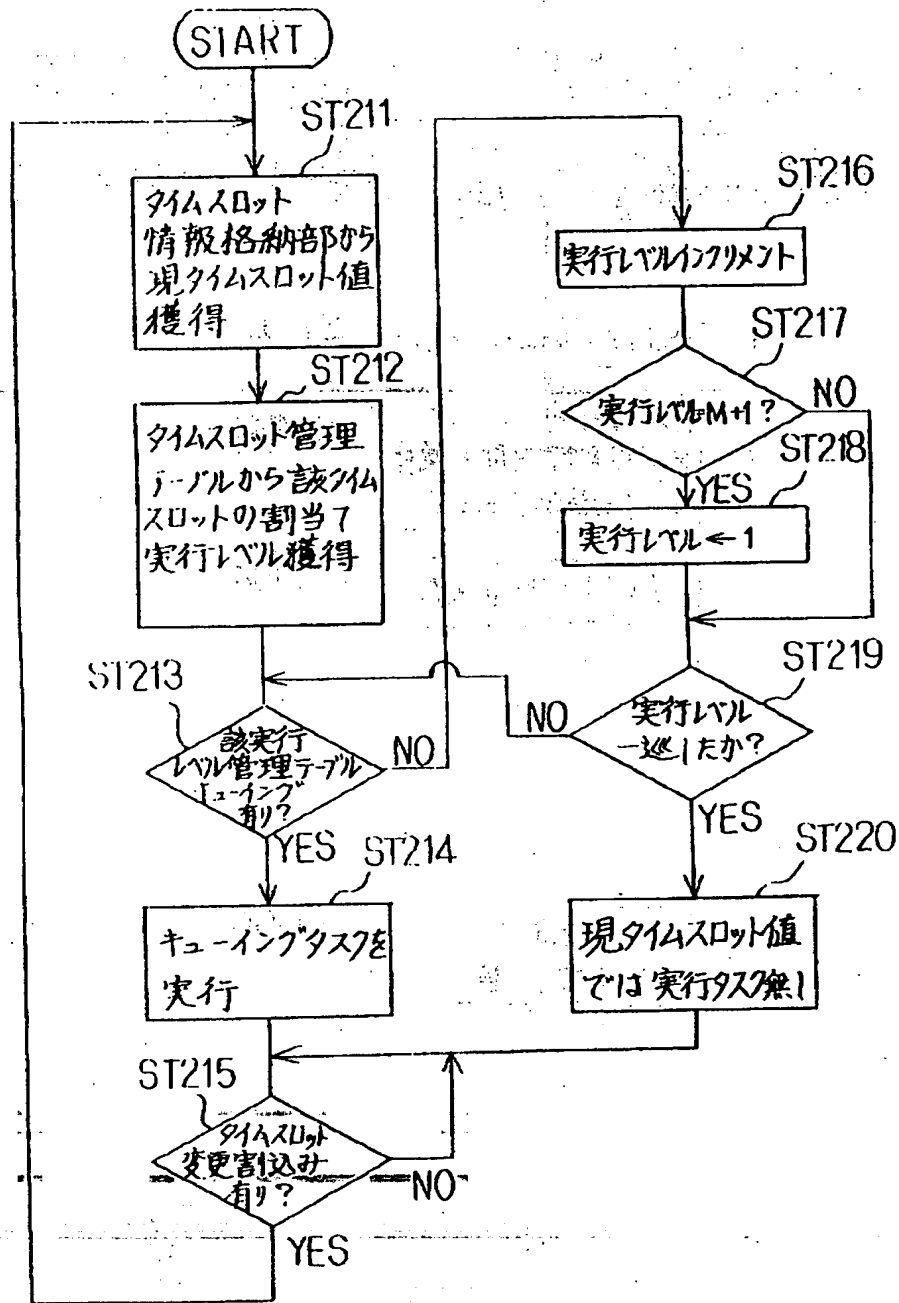
【図18】



【図19】



【図21】



【手続補正書】

【提出日】平成5年5月20日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0003

【補正方法】変更

【補正内容】

【0003】まず、周期的スケジューリングの概念につ

いて、図16を用いて説明する。図中の最小基準周期とは、周期的にタスクの起動をチェックする際の基準周期となるもので、周期的に実行されるアプリケーションタスク1～L(Lはタスク数)をスケジューリングする最小周期である。タイムスロットとは、上記最小基準周期を所定値N(ユーザがシステムに合わせて決定する)で割った時間単位で、タイムスロット時間毎に、スケジュー

ーラが実行タスク1～Lの選択を行う。実行レベルとは、実行タスク1～Lを周期毎に分類した複数のタスクの集合で、タイムスロット1～Nにその実行レベル1～Mが割当てられる(割当て方法は後述する)。スケジューラはスケジューリング実行時に、あるタイムスロットでそのタイムスロットに割当てられた該実行レベルの起動待ちタスクを優先的に実行する(実行レベルの選択方法は後述する)。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0004

【補正方法】変更

【補正内容】

【0004】図16の例では、タイムスロット1とタイムスロット2に実行レベル1を、タイムスロット3に実行レベル2を、タイムスロット(N-1)に実行レベルMを割当て、タイムスロットNは、実行レベルの選択時間やタイムスロットの計算時間として、各タイムスロット処理に要するシステム時間のための予備のタイムスロット(α)であり、実行レベル1を割当てる。実行レベル1にはタスク1とタスク2の2つのタスクが、実行レベル2にはタスク3が、実行レベルMにはタスクLが所属している。スケジューリング時、タイマの値から、後述するタイムスロット計算部がタイムスロット値を計算する。最小基準周期毎に、実行周期が巡ってきた実行レベルのタスクが後述する起動タスクキューイング部により起動要求されるが、異なった実行レベルが同時に起動要求された場合、以下の様にタイムスロット毎に起動要求実行レベルの切替が実行される。タイムスロット1においては、実行レベル1が選択され、実行レベル1のタスク1が実行されている。タイムスロット2においても、実行レベル1が選択され、同様にタスク1が実行されている。タイムスロット3においては実行レベル2が選択され、実行レベル2の所属タスク3が実行される。タイムスロットNは、予備のタイムスロットなので、最小基準周期内で未実行のタスクが実行される。以上の様な実行レベル選択が、最小基準周期毎に繰り返される。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0020

【補正方法】変更

【補正内容】

【0020】タイムスロット割当て後のスケジューリング実行は、従来の手段と同様の動作を行う。この時、プライオリティスケジューリングモードで図5に示したようにタイムスロット管理テーブル211が生成された場合は、従来例で示した図21のフロー図は図6に示すようになり、全てのタイムスロットで実行レベル1を得て(ST221)、実行レベル1から図22に従ってレベル管理テーブル212a～212cをチェックし、最初

に発見したキューイングタスクを実行する。この結果、実行レベル1に最大のプライオリティを与え、実行レベルMに最小のプライオリティを与えることになる。即ち、どのタイムスロットにおいても実行レベル1を獲得し、図22の実行レベル探索に従い実行タスクを決定するので、実行レベルの小さいタスクが終了しない限り、大きいタスクは実行されない。この結果、実行レベルの小さいタスクほど高いプライオリティを与えるプライオリティスケジューリングが実現される。この場合、従来例で示した図16は図7に示ようになる。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0025

【補正方法】変更

【補正内容】

【0025】次に、本実施例のスケジューリング動作について、図14のフロー図を用いて説明する。H/Wタイマ34のクロック36を用い、割込み発生器35はタイムスロット時間毎にスケジューリング実行部13に割込み37を発生する。カウンタ(1)38からカウンタ(N)40はH/Wタイマ34のクロック36を利用して、それぞれ定められたタイムスロット値1～Nをカウントし、最小基準周期毎にタイムスロットレジスタ41の対応ビットをセットする。スケジューリング実行部13は、割込み発生器35からタイムスロット時間毎の割込み37が入るとタイムスロットレジスタ41を見に行き、セットされたビットからタイムスロット値43を得る(ST141)。スケジューリング実行部13は、タイムスロット値を獲得後、タイムスロットレジスタ41をリセットする。このタイムスロット値を用いて、実施例1と同様のスケジューリング処理を行ない、H/Wタイマ割込みを待つ(ST145)、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。タイマにH/Wタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。即ち、実施例1では、オペレーティングシステムのS/Wタイマ機能を用い、現在の時刻を獲得する処理、タイムスロットの計算、スケジューリング実行部13へのタイムスロット変更割込みの通知といった処理を全てソフトウェアで実現している。この場合、タイムスロットの前回の保持が必要な上、オーバーヘッド時間(図7のシステム時間)が大きくなるという欠点がある。これに対し、本実施例のように、以上の機能を、H/Wタイマを用いて割込み発生器、カウンタ及びレジスタなどのハードウェアで実現した場合、タイムスロットの前回の保持が不要となる上、オーバーヘッドを低減でき、高速なスケジューリングが可能となる。

【手続補正5】

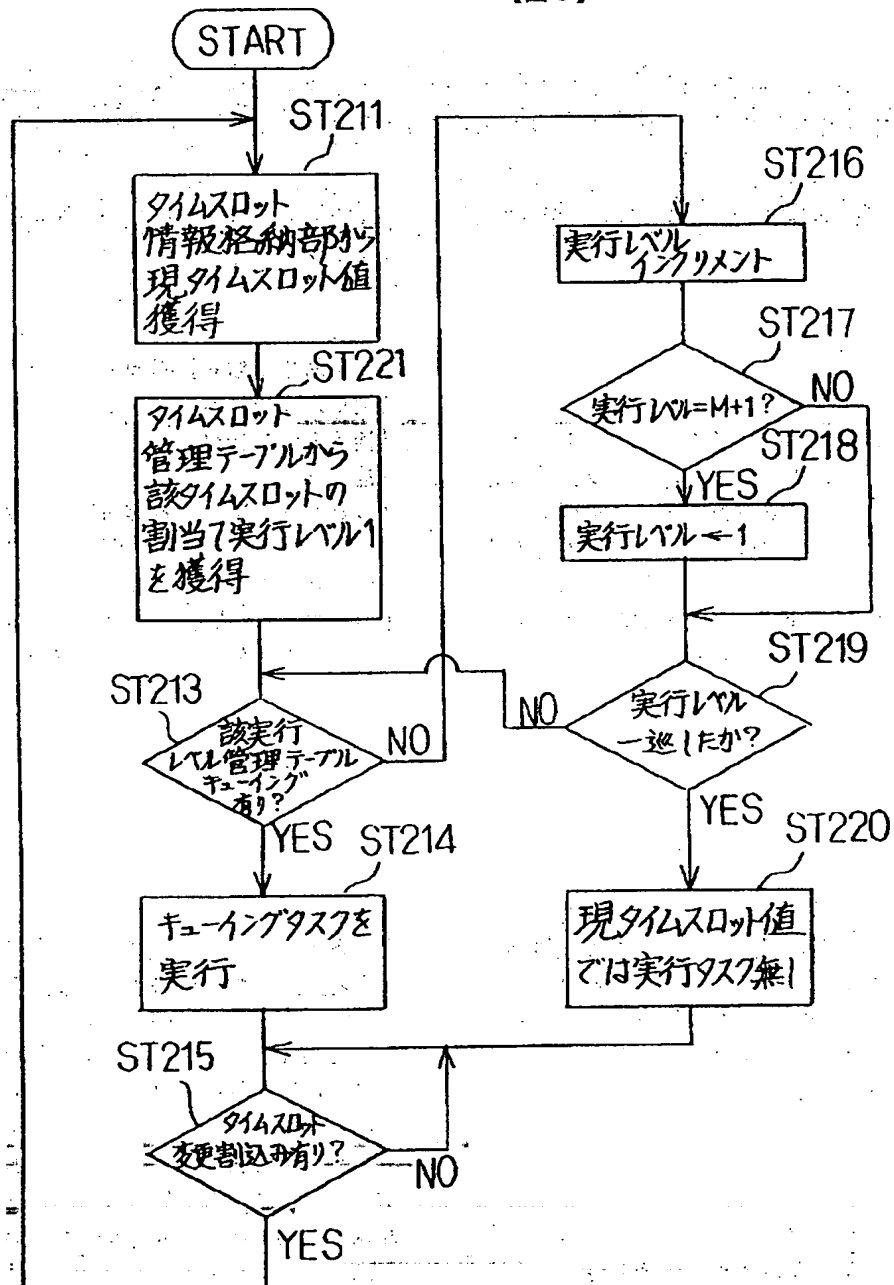
【補正対象書類名】図面

【補正対象項目名】図6

【補正方法】変更

* 【補正内容】

* 【図6】



【手続補正6】

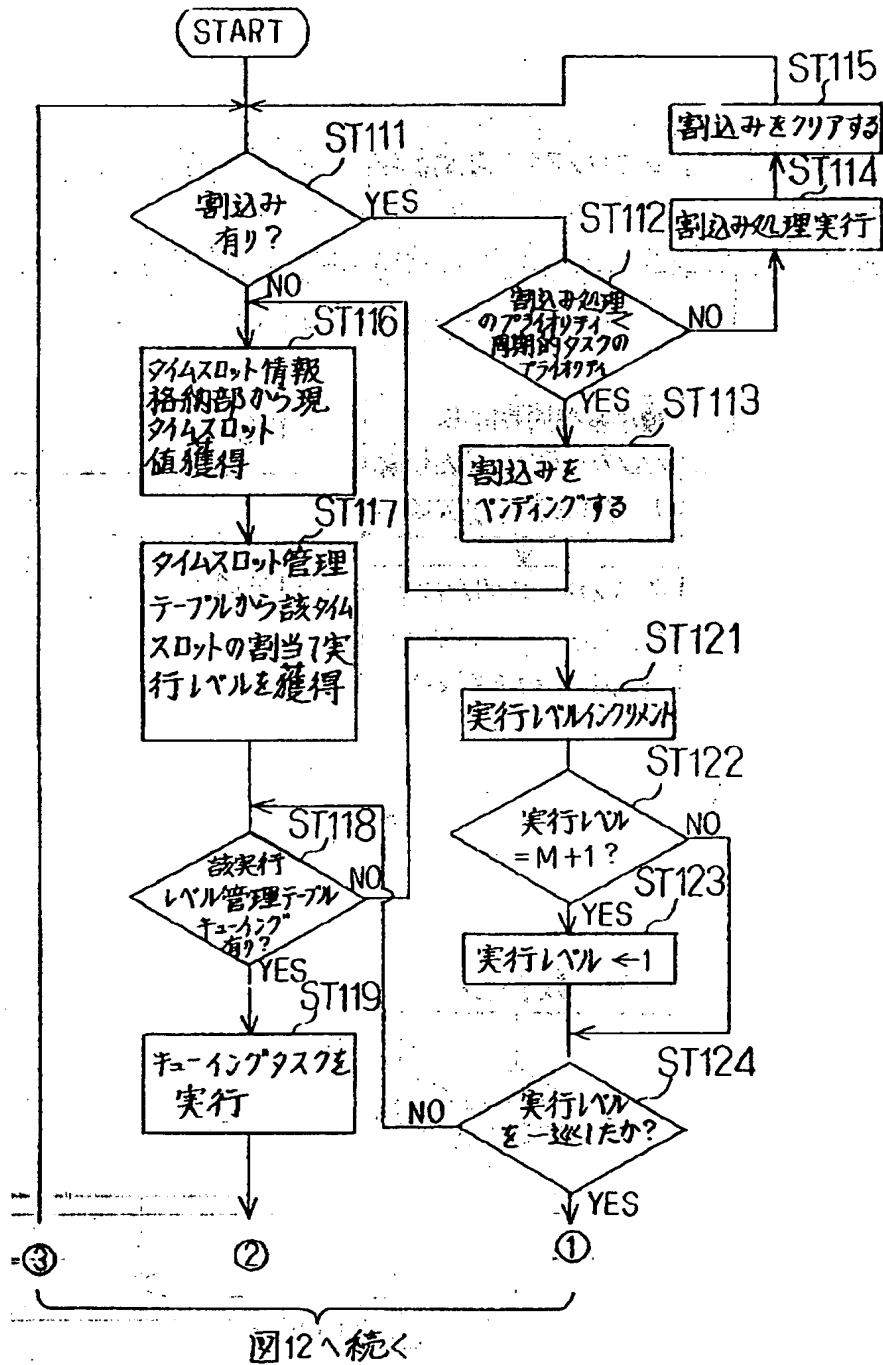
【補正対象書類名】図面

【補正対象項目名】図11

【補正方法】変更

【補正内容】

【図11】



【手続補正7】

【補正対象書類名】図面

【補正対象項目名】図19

【補正方法】変更

【補正内容】

【図19】

